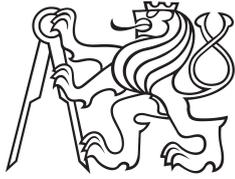


Bachelor Thesis



**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Cybernetics**

Exploratory action selection to learn object properties through robot manipulation

Andrej Kružliak

Supervisor: Mgr. Matěj Hoffmann, Ph.D.

Supervisor–specialist: Jan Kristof Behrens, MSc.

Field of study: Cybernetics and Robotics

May 2021

Acknowledgements

I would like to thank my supervisor Matěj Hoffmann for his guidance and support, whether professional or emotional. I would also like to thank Jan Kristof Behrens for the guidance in ROS, and help with the simulation, Shubhan Patni for help provided with data processing, Ville Kyrki and other IPALM members for expert theoretical guidance and feedback, my family and last but not least, my high-school science teacher, thanks to whom I went to university in the first place.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May , 2021

Prehlasujem, že som predloženú prácu vypracoval samostatne a že som uviedol všetky použité informačné zdroje v súlade s Metodickým pokynom o dodržiavaní etických princípov pri príprave vysokoškolských záverečných prácí.

V Prahe, máj, 2021

Abstract

Action selection is a term used to describe a process of autonomous selection of steps that lead an agent to a predetermined goal. In this work, discovering object properties and the overall object category (e.g., material, or box, mug, etc.) by a robot manipulator is the desired goal. Extracting properties of objects from visual input only is limited, especially regarding physical/material properties like surface roughness, stiffness, or mass. Here, haptic exploration, i.e., mainly proprioceptive and tactile input during manipulation of the object, is indispensable. Furthermore, unlike visual sensing, which is often passive—images taken by a static camera—haptic exploration is intrinsically active: the particular way of manipulating the object determines the quality of information that can be acquired. Here, this idea is formalized, and robot actions (compressing or lifting objects) are assessed by how much they are likely to reduce uncertainty about specific object properties—their expected information gain. The most informative action is then chosen. The expected information gain is calculated in three different modes based on information entropy, which is estimated for both discrete probability distribution of material composition of the object (e.g., plastic, ceramics, metal) and continuous distribution of each property like elasticity or density. We use classification as a proxy metric of how optimal are the choices of the action selection algorithm. Overall the mode optimizing for the information gain of the continuous properties results in the best classification. Learning of object properties is accomplished in the form of a Bayesian update from real measurement actions. Such selection of actions leads to more efficient learning about the environment and, as a result, helps the agent in navigating the real world, where the unexpected shall be expected.

Keywords: haptic object exploration, robot manipulation, robot grasping, action selection, information gain, information entropy

Supervisor: Mgr. Matěj Hoffmann, Ph.D.

Abstrakt

Výber prieskumných akcií je pojem popisujúci proces autonómnej selekcie krokov, ktoré vedú agenta k predurčenému cieľu. V tejto práci, je cieľom skúmanie vlastností a celkovo kategórie daného objektu (napr. materiál, krabica, šálka a pod.) robotickým manipulátorom. Extrahovať vlastnosti objektu len vizuálne je limitujúce, najmä v spojitosti s fyzikálnymi/materiálnymi vlastnosťami ako povrchové trenie, tuhosť, či hmotnosť. V rámci tejto práce je hlavným interaktívnym prvkom dotyk, teda najviac informácií je získavaných z haptickej manipulácie predmetom. Narozdiel od vizuálnych vnemov, ktoré sú pasívne—fotografie zaobstarané statickou kamerou—haptické skúmanie je v samotnej podstate aktívne: spôsob manipulácie priamo ovplyvňuje množstvo informácií, ktoré je možné získať. V tejto práci je táto idea sformalizovaná, kde sú volené ďalšie robotické akcie (stláčanie, či dvíhanie objektov) na základe toho, ako je pravdepodobné, že na základe danej akcie príde k zníženiu neistoty v rámci vlastností—teda na základe ich očakávaného informačného zisku. Akcia, ktorá prináša informácií najviac, je zvolená. Očakávaný informačný zisk je počítaný v troch rôznych módoch založených na informačnej entropii. Informačná entropia je odhadovaná ako pre diskkrétne pravdepodobnostné rozdelenie materiálovej kategórie, tak i pre spojité pravdepodobnostné rozdelenie vlastností, ako pružnosť, či hustota. Používame klasifikáciu ako proxy metriku toho, ako veľmi sú rozhodnutia algoritmu ohľadne selekcie akcií optimálne. Mód optimalizujúci pre informačný zisk spojitej premennej vykazuje najlepšie výsledky. Učenie sa vlastností objektov je zabezpečené pomocou Bayesovskej aktualizácie z meraní priamo manipulátorom. Takýto výber akcií vedie k viac efektívnemu učenie o okolí a ako výsledok pomáha agentovi v navigácii reál-

nym svetom, kde je potrebné očakávať aj neočakávané.

Kľúčové slová: haptické skúmanie objektov, robotická manipulácia, robotické uchopenie, výber akcie, informačný zisk, informačná entropia

Preklad názvu: Výber prieskumných akcií za účelom zistenia vlastností predmetov robotickou manipuláciou

Contents

1 Introduction	1	Bibliography	55
1.1 Motivation	1	Project Specification	59
1.2 Goals	2	A Action selection – Virtual	61
2 Related Work	3		
3 Materials and Methods	5		
3.1 Experimental setup	5		
3.1.1 Theoretical model	5		
3.1.2 Physics Engine – MuJoCo	6		
3.1.3 Simulation setup	8		
3.1.4 Hardware setup – Kinova Gen3, Robotiq 2F-85 and Object properties	13		
3.2 Probabilistic reasoning	15		
3.2.1 Bayesian network	16		
3.2.2 Bayes theorem	18		
3.2.3 Bayesian inference	18		
3.2.4 Example of Bayesian inference	19		
3.2.5 Obtaining a prior	22		
3.3 Action selection	23		
3.3.1 Information theory	23		
3.3.2 Entropy and differential entropy	24		
3.3.3 Action selection algorithm	24		
3.3.4 Differential entropy of a property belief	25		
3.3.5 Material entropy	26		
3.3.6 Differential entropy of a property with relation to a property emulation	27		
3.3.7 Bayesian Message Passing	29		
3.3.8 The Action selection algorithm	30		
3.3.9 Measurement uncertainty and a placeholder category	33		
3.3.10 Connection to IPALM database	35		
4 Experiments and Results	37		
4.1 Virtual Experiments	38		
4.2 Simulation Experiments	40		
4.3 Real-World Experiments	44		
4.3.1 Results	47		
5 Conclusion	49		
6 Discussion and Future Work	51		
6.1 Future Work	52		

Chapter 1

Introduction

1.1 Motivation

Since the industrial revolution, machines and other mechanical devices helped to significantly raise the margins of production. Many processes could be suddenly parallelized and later automated. Growing technological progress brought more precise and vastly more efficient forms of construction, welding, manipulation and much more, in the form of robotic manipulators.

The automation started in the industrial branches with highest demands for efficiency and precision, such as automotive industry, microelectronics, aviation and so on. Automation, heretofore a concept exclusively reserved for hi-tech industries, became a part of the everyday, modern life. It takes place in functioning and maintenance of stores, warehouses, households, transportation and many more nuances of our everyday lives. The presence of robotics switched from laboratory and factory exclusive to omnipresent. This omnipresence of progressively competent and relatively intelligent machines creates new robotic related decision-making and perceptive challenges. The robotics are already an established part of the automated industry, but their applicability in everyday-life situations in an open and uncontrolled environment is still not given. Such application requires integration of previous experiences into the reasoning, successful interaction with the physical world and also navigating the world, which was built by humans—for humans.

An average educated human, if presented with an object, is able to recognise its shape, estimate the material properties and reasonably assess its weight and/or center of mass from visual observation and/or haptic experience [1, 2]. That is, after all, the everyday reality of being a human. However, to minimize the uncertainty about their beliefs, they may also choose to interact more intensely with the object. Grasping, lifting (relative weighing), rubbing with fingers against its surface and so on. The more the person explores and consequently learns about the object, the more confidence can be put into the object categorization and therefore manipulation and utilization. This thesis aims to tackle the challenges related to planning of the exploration and learning from the environment with a robotic manipulator, whilst getting

inspired from the physical object-recognition routines of humans. This work is a part of a collaborative project *Interactive Perception-Action Learning for Modelling Objects* (IPALM) between five academic institutions in Europe [3]. The project's purpose is to “develop methods for the automatic digitization of objects and their physical properties by exploratory manipulations”.

■ 1.2 Goals

The goal of this thesis is to create and formalize a framework for selecting appropriate actions for better understanding of the inspected object. The framework should provide a metric to evaluate the information gained from previous actions. Based on the information gathered to that point, an educated prediction could be expressed, possibly predicting how much learning is expected to happen in any of the next possible actions. Based on this metric, algorithmic approach should be utilized for choosing the action, which promises the largest quantity of learning - information gain. This whole framework should be constructed in a simulation and real life, to create a proof of concept. As stated before, this thesis is part of IPALM project [3], so a physical realization of this information gathering pipeline is desired.



Chapter 2

Related Work

An action as primitive as grasping an object in the context of robotic manipulation can often prove to be difficult. This action often requires either a very precise calibration of the robotic embodiment or an iterative approach. Once the grasp of the object with a robotic manipulator is possible, the grasp position, force and other parameters are often tailored to the particular manipulator. Transfer of this knowledge (from concrete embodiment to an abstract representation) is tackled in [4] by Felipe et al. Many corrective practices are deployed (e.g. small variable movements in the grasping procedures). This allows for a *blind exploration strategy*, where an arm follows a set exploratory trajectory until contact is detected. This allows for a grasp to be planned and executed. Learning about the object properties was out of the scope of the work mentioned and this thesis offers a possible follow-up in the whole process. The algorithm proposed in this thesis is primitive-independent. Primitive-independent means that the algorithm proposed in this thesis does not rely on any specific action, it only relies on the information gained from such action (e.g. measurement mean and deviation).

Liarokapis et al. [5] proposed grasping techniques for underactuated grippers that could discriminate between different objects using only a single force closure grasp with force sensors. This underactuated sensing does not provide complex understanding of the object's pose and shape, but is on the other hand able to recognize objects by training a random forest classifier and is model-free. So the only data used for discrimination is the raw data from the sensors. Interestingly, accuracy of the recognition did reportedly not drop, while reducing the number of force sensors from 8 per finger to 2 per finger. The compliance of the robotic hand used in the paper may be an obstacle in future deployment of such approach in non-compliant grippers.

Attempt to grasp an object might be done with high precision in industrial environment, although to be able to successfully grasp an object in general conditions with limited knowledge and great uncertainty, a more complex solution is needed. Nikandrova et al. [6] suggested an explorative extension to the general grasping procedure. The cited paper focuses on maximizing the stability of a grasp, while including also an exploratory sequence which helped to learn needed information about the object, if necessary. This allowed for many more successful grasps despite of having either incorrect prior knowledge

or a great measurement uncertainty. The initial grasping, similarly to this thesis, is chosen via Bayesian inference based on the knowledge possessed before the time of the first grasp attempt. The grasp with the maximal expected stability is chosen. If this grasp was not expected to be stable with relation to some threshold, exploratory action was chosen. This action searches for the exact position of the explored object. A multivariate entropy function regarding the object's position is constructed and a minimum of this landscape is found utilizing the MCMC (Monte Carlo Markov Chain) method. The most stable grasp of that minimum is executed as an expected most informative grasp and the information on location is updated.

In this thesis, MCMC methods are not needed, the entropy function is univariate in every property and the shape is well understood.

Whether for just the proof of concept or an actual machine learning implementation, simulations might play a significant role in the learning about the objects or about the expected informativeness of the grasps. The capabilities of physics engine are quite welcome in most of the grasping simulations and most commonly, MuJoCo and Gazebo are the ones used in such applications [7]. MuJoCo simulator is also used to show some proofs of concept in this thesis.

McGovern et al. [8] explored learning to estimate the center of mass (CoM) of arbitrary objects. This was done by Q-learning in a simulation environment Gazebo, which like MuJoCo, supports physics simulation. That was necessary for the experimental learning. The CoM was firstly assumed to be the center of the bounding box of the assessed object, which was incrementally (in small steps) pushed to the edge of testing surface. This was iterated until the CoM component in the explored direction was found with the demanded precision. This procedure was repeated in all three axes and provided rewards for the Q-learning algorithm to adjust weights in a neural network for faster and more precise CoM assessment in the future.

Haptic exploration of the environment mostly relates to object detection, pose recognition, elasticity estimation and so on. Nguyen Le et al. [9] focused mainly on extracting friction information from arbitrary objects with the combined use of vision and haptic exploration, which adds yet another exploratory action to the possible toolkit. Friction estimation is not only useful for object recognition and classification, but also provides more efficient grasps in the regions with higher friction. A probabilistic model is used to predict the friction coefficients on the whole object. A prior from vision point-cloud is filtered and segmented. Segments are explored via touch and the prior and measurement information is combined to create a variable friction model. The output of the model is a friction map and a confidence map. Their proposed method does not make the assumption of homogenous friction which as a result helps with grasping in the regions of highest friction density.

Chapter 3

Materials and Methods

All the code used in this thesis is available at the dedicated Gitlab online repository [10]. Complementary visual and audio-visual content is available at the dedicated online Google Drive [11].

3.1 Experimental setup

In this section a description of different tools and approaches used for experimenting with action–selection is provided. Firstly, it is important to lay the foundation of the working environment, which consists of object categories and object properties. In order to navigate through this space a way of interaction with the object possessing the category and properties is needed. That is done based on theoretical evaluation of the current probabilistic belief in property and category and acted out in the simulation and/or the physical world. A visualization of the pipeline can be seen at the end of this section, in the Fig. 3.28.

3.1.1 Theoretical model

The whole action selection process will be held in a variable space with only four known materials. This simplification serves practical purposes, as the functionality of the proposed action selection algorithm does not significantly change with scale. Regarding the material properties, it is reasonable to expect the variables to be normally distributed around the expected value. The same assumption is made with measurements, which are assumed to be normally distributed with a mean equal to the measured value and standard deviation equal to the measurement error. Reference values were approximately chosen from different sources [12, 13, 14, 15]. It is not important for this thesis to prudently choose the parameters for each material, since it would be equally useful to test the action selection on arbitrary materials A , B , C and D . The choice of materials and their real world approximate equivalents is useful for good intuition building. The standard deviations σ_{REF} for materials were chosen to account for different alloys and slightly vague material definitions, such as *plastic* and *ceramic*, where the name represents a wider group of materials.

Material	Density [kg m ⁻³]		Elasticity [GPa]	
Parameter	μ_{ref}	σ_{ref}	μ_{ref}	σ_{ref}
Steel	7900	250	200	10
Ceramic	2300	50	30	10
Plastic	1350	250	3	10
Aluminium	2705	180	70	20

Table 3.1: Table of reference materials and their normal parameters.

The table 3.1 consists of two types of material properties, that are of interest in this thesis. The elasticity will be referred to via Young’s modulus. In short, Young’s modulus is a constant of proportionality which puts into perspective how does the change of length relate to the cross-sectional area, when pushed or pulled with a force. The amount of stretch is proportional to the original length and inversely proportional to the cross-sectional area [15, Ch. 9.5] This is written as

$$E = \frac{\text{stress}}{\text{strain}} = \frac{F/A}{\Delta L/L_0}, \quad (3.1)$$

where *stress* is the pressure applied (force F per unit area A) and strain is described as the change in length ΔL to the original length L_0 . The units for Young’s modulus are *Pascals* [Pa]. The action selection will be made from the action space defined as follows:

1. **Observe** - The vision pipeline, provides material category prior; in the future may also provide shape information.
2. **Grasp** - As trivial as it may seem, grasping of a complex object, such as a mug, is difficult without the shape information. Alternatively, grasp attempts may be used to learn about the object shape, which is left for further work. Grasping is a prerequisite for lifting.
3. **Squeeze** - This action may need **Grasp** as a prerequisite, if the squeezing is done with a gripper, opposed to a piston or a plate. The action provides information about the elasticity.
4. **Lift** - Lifting is useful for transportation and possibly the estimation of the center of gravity. This action is a prerequisite for weighing.
5. **Weigh** - Weighing may be done in different configurations, this will be elaborated later. Torque sensors are useful for this application.
6. **Rub** - This action is not yet available, but is being developed by other participants from the IPALM project [9].

3.1.2 Physics Engine – MuJoCo

MuJoCo is a physics engine used for research, robotics development, machine learning and many other disciplines with the need for physics. MuJoCo

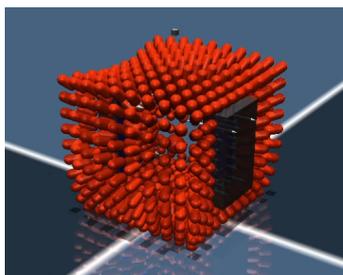
is one of the few physics simulators with a Soft-Body contact simulation available which makes it the simulator of choice in many robotics applications [16]. In fact, for robotics, it is the second most cited right after the Gazebo simulator, which does not provide Soft-Body contact simulation. The name is an abbreviation for *Multi-Joint dynamics with Contact*. In this work, MuJoCo is used mainly for the Soft-Body simulation. The engine is also later used to simulate the control pipeline of the Kinova manipulator, which can be later used similarly for the actual physical Kinova setup.

■ Squeezing: 2F-85 gripper imitation

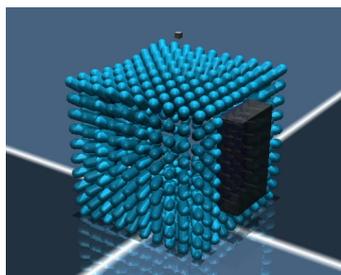
To gather some simulation data for future reference for either simulation or real-world application a squeezing setup was designed. Soft objects in four different categories (*very soft*, *soft*, *medium* and *hard*) were created. The MuJoCo Soft-Body simulation is not based on real world proxies, such as Poisson's constant, Young's modulus or Hooke's law, but rather focuses on the particle-wise relationships of the particles in the soft object. This is shown in a minimal example code Listing 1. An illustration of the Soft-Body properties of such objects can be seen in the Fig. The basics of composite Soft-Body object creation in MuJoCo are based on the bachelor's project by Michal Pliska [17]. 3.1.

```
<!-- Very soft -->
<composite type="box" count="10 10 10" spacing="0.0065">
  <geom type="capsule" size=".002 .002" rgba="1 0.15 0 1" mass="0.003"/>
  <joint kind="main" stiffness="1" damping="1"/>
  <tendon kind="main" stiffness="1" damping="1"/>
</composite>
```

Listing 1: Soft object minimal example XML code for MuJoCo simulation.



(a) : Squeezing of a Very Soft cube.



(b) : Squeezing of a Hard cube.

Figure 3.1: Squeezing with a 2F-85 gripper imitation in MuJoCo.

Four materials of different `stiffness` levels were used. The trajectory was also varied with different end position goals: -1 and -2 cm. In other words, each finger was trying to push its way to the other side through the cube. The movement was obstructed by the soft object and the collision resulted

in force readings on the force sensors located on the fingers. The variation in the end position goals mimics the gripper effort. An example of the data gathered in this process are visualized in the Fig. 3.3.

The MuJoCo engine provides great possibilities for simulating motion of physical objects and their collisions. While the simulation works great at small speeds and small forces, the collisions must be computed for each time step. This brings a lot of computational load and with great forces, the gripper simply slips through the cube. This may be caused by the discrete nature of the soft object, which is a composite of many capsules connected with tendons. Those tendons have some stiffness, which can be either overcome if set to low values (small stiffness), or will crash the simulation if set to high values (large stiffness) and pressed upon with great force. Such “numerical penetration” can be seen in the Fig. 3.2. More details on MuJoCo usage can be found in [18].

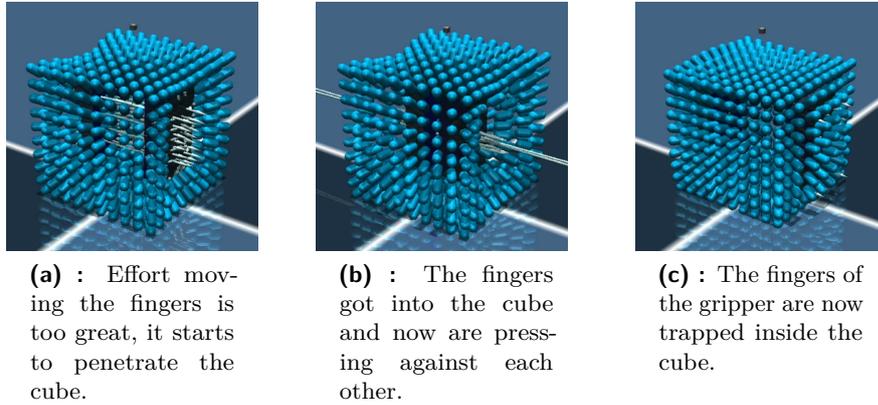


Figure 3.2: Squeezing with a 2F-85 gripper imitation in MuJoCo.

The Subfig. 3.2(b) shows some white poles coming outward from the cube. Those white poles are, in the full picture, the visualisation arrows of the resulting contact forces. This is caused by the actual contact of the two fingers pressing against each other. The plates used for the squeezing simulate the size and overall shape of the *Robotiq 2F-85* gripper used on the real manipulator.

■ 3.1.3 Simulation setup

In this section a more complex description of the simulation setup is provided. In contrast with the previous section, where the squeezing was rudimentary, here the whole manipulator is incorporated. The physics engine does not only simulate the Soft-Body and two moving plates, but also needs to simulate the dynamics of the whole manipulator. A Soft-Body cube is presented to the manipulator. The manipulator then selects an action based on prior knowledge and executes it. This can be either weighing, or squeezing as seen on the Fig. 3.4.

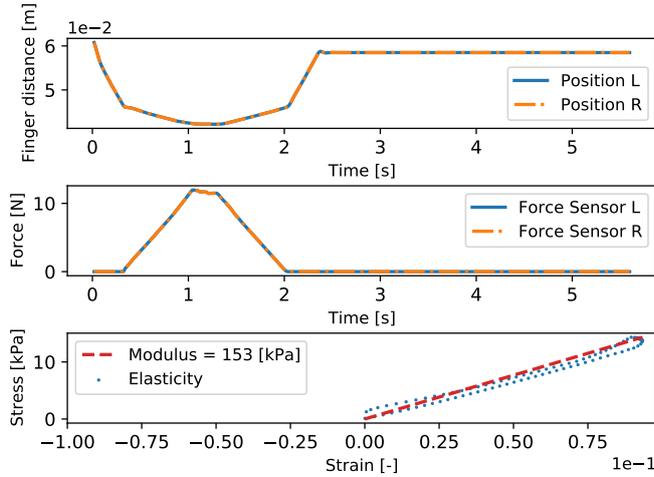


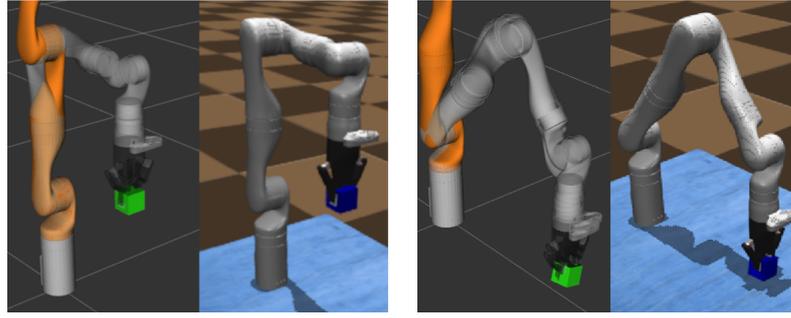
Figure 3.3: Example of the time series data from the 2F-85 gripper imitation sensors (top and middle) and a Stress-Strain curve (bottom) with the Young’s modulus in the legend, of the Hard cube squeezing.

Despite the previous section showing high refresh-rate and high resolution measurement, a more complex simulation pipeline inhibits the performance. That is because many more parts are simulated at once. The pipeline consists of Robot Operating System (ROS) and the MuJoCo simulator itself. ROS is a versatile open-source middleware operating system. A big collection of community contributed code is available for the development of new robotic systems. Applications in ROS are composed of many concurrently running programs, the nodes. For the communication is between the nodes, ROS implements interprocess communication patterns publisher/subscriber, server/service and actions. It provides a good overview of the system’s control by tools like RViz and RQt, hardware abstraction and implements messaging protocols for simple and reliable communication. It is generally multipurpose, but mainly used for research robotics, drones, humanoid robots etc. [19, p. 55, p. 147]

In this work, MoveIt motion planning framework for ROS is utilized, commanding the Kinova Gen3 manipulator inside of the MuJoCo simulation. An essential part of this pipeline, connecting ROS and MuJoCo was provided by my supervisor specialist *Jan Kristof Behrens, MSc* [20]. Building on the provided core, I was able to connect the later described action selection algorithm and the robot and proceed with designing the exploratory actions.

■ Density estimation via Weighing action

One way to learn more about the object is to assess its weight. Although, considering that one tonne of feathers and one tonne of steel weigh, in fact, the exact same amount, learning only about the weight is not alone sufficient.



(a) : Weighing action portrayed in ROS visualization interface RViz (left) and in MuJoCo (right).
 (b) : Squeezing action portrayed in ROS visualization interface RViz (left) and in MuJoCo (right) cube.

Figure 3.4: Visualization of actions in MuJoCo simulation.

The weighing action may be used to estimate the object's average density, which also needs the information about the shape, or to be more precise, the volume. For this thesis, a simplifying assumption is made, that the measured objects are cubes and their volumes are known. Knowing density has greater value than knowing the mass alone, because it can be used for object classification, or a more valuable learning / exploratory stimulus. The mass is estimated based on output from the robot's joint torque sensors. To minimize the measurement error in torque reading, all possible robot axes are aligned with the direction of the gravitational vector \vec{g} . The torque is

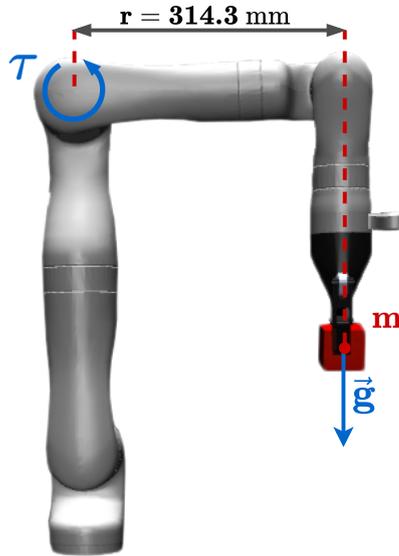


Figure 3.5: Diagram of the weighing process.

calculated as

$$\vec{\tau} = \vec{r} \times \vec{F}, \quad (3.2)$$

where \vec{r} is the lever and \vec{F} is the force acting on the lever. In the Fig. 3.5, it can be seen, that the whole measurement process lies in one plane. Therefore we can write

$$\tau = \|\vec{r}\| \cdot \|\vec{F}\| \cdot \sin \theta, \quad (3.3)$$

where θ is the angle between the lever and the force. The force acting on the lever is the gravitational force $\vec{F}_g = m\vec{g}$ and the elbow (the fourth joint) and the wrist (the sixth joint) are both in $\pi/2$ configuration. This effectively means that the gravitational force is perpendicular to the arm connecting the wrist and the elbow and parallel with the axis of the forearm, which is the segment connecting the wrist and the end effector. Therefore we can write

$$\tau = r \cdot m \cdot \vec{g}. \quad (3.4)$$

The equation may be rearranged to yield the mass as

$$m = \frac{(\tau - \tau_0)}{r \cdot \vec{g}}, \quad (3.5)$$

where τ_0 is the holding torque of the empty gripper. This value is calibrated at the beginning of the measurement.

■ Young’s modulus estimation via Squeezing action

Another way to learn additional information about the object’s properties is squeezing or pressing. Similarly as with density, Young’s modulus is a measure of the material’s intrinsic properties, which help with either object classification or exploration. The process of estimating the modulus was described in the previous sections, namely Eq. 3.1. An example of data gathered directly from the ROS–MuJoCo simulation can be seen in Fig. 3.6. Although simulation with the whole pipeline opposed to the bare physics of previous Section 3.1.2 brings problems.

The model of Kinova Gen3 manipulator is described mainly in polygons (meshes) and the physics of the contacts are handled by MuJoCo. Although the immediate contact area of the gripper and the Soft-Body is a MuJoCo Box object, the surrounding gripper and manipulator components are constructed with meshes. This probably brings unwanted contacts while over-squeezing and creates too many contact points, which results in shaking and sometimes erratic behavior reminiscent of an explosion. What adds to the instability, is simulating the friction. To successfully grasp and lift an object, an indispensable amount of friction must be generated between the gripper and the cube to resist falling. This has been stabilized by choosing the same solver-constraint dimensions for friction in both object and the gripper.

Yet another limiting factor is the number of Soft-Body capsules per cube facet. The amount of capsules dictates the resolution with which the solver registers the collisions. There is a limit balancing the number of capsules per facet size which dictates how strongly may the object be pressed with the gripper, before unwanted penetration occurs. Such measurement with

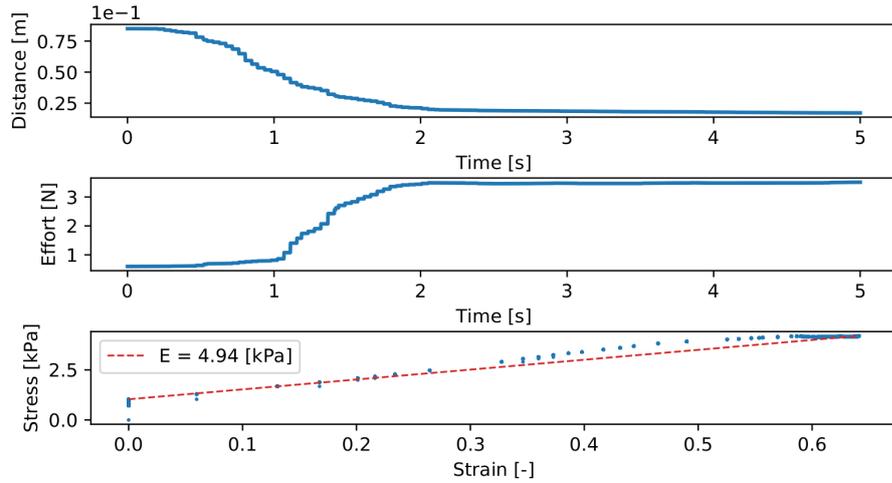


Figure 3.6: Example of the time series data from the squeezing of an arbitrary soft object with 2F-85 gripper, obtained from the ROS-MuJoCo simulation.

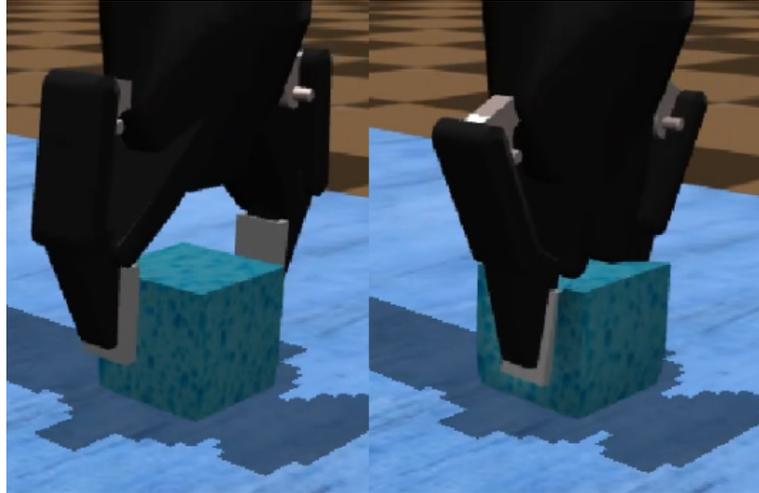


Figure 3.7: Detail on the deformation of a cube in MuJoCo-ROS simulation.

penetration is unusable. Example of unusable measured data is shown in the Fig. 3.8. This considerably limits the scale of stiffness, which is available for direct squeezing. The exact cause of the difference between the bare MuJoCo pressing and the more realistic squeezing with the Robotiq gripper is unknown. It is suspected that the gripper translates all of the minute movements and jitters of the whole manipulator which reflects on the robustness of every movement. Another cause may be the fact that the bare MuJoCo pressing is done in a linear motion, whereas the gripper squeezes in circular motion allowing for some parts of the gripper to touch the cube sooner than the rest.

Creating a reference database for the squeezing comes with a challenge. When measuring the squeezing data to create the database, a mass of 1 kg was set to each measured cube. Five measurements were done for each cube

and the average of the Young’s moduli was used as a referential value for the database. Although when changing the mass of the cubes for experiment purposes, the change in mass also slightly reflects in the Young’s modulus. It is not clearly known, why this happens, but there exist a few possible explanations. The physics engine also simulates the effects of gravity. The Soft-Body is simulated by a network of joints and tendons. When the gravity presses on the cube, it is generating pressure. The tendons are compressed and make the cube pre-loaded. In this scenario, the gripper needs to not only overcome the natural stiffness of the cube, but also the pre-loaded pressure from the tendons. Another possible explanation is that with greater mass, the inertia of each capsule is larger. Therefore the capsules resist motion to a greater extent. The phenomenon of slightly variable elasticity biases the measurement, based on how far is the current mass from the mass used to measure reference.

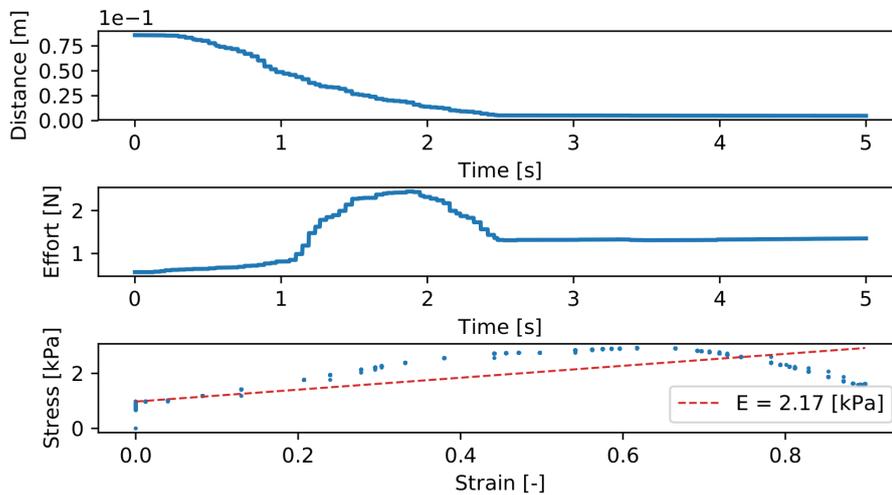


Figure 3.8: Exemplary time series record of a penetration through the cube’s faucet with the 2F-85 gripper imitation in MuJoCo. The Effort curve is resembling a hill shape. The effort drops, when penetration occurs.

3.1.4 Hardware setup – Kinova Gen3, Robotiq 2F-85 and Object properties

For the actual manipulating and overall interacting with the explored object a Kinova Gen 3 manipulator is utilized, which provides 7 Degrees of Freedom. It is an ultra-weight robot not suited for industrial settings [21]. Its dimensions can be seen in the Fig. 3.9. Kinova also comes with its own API and ROS packages, called `Kinova ROS Kortex`, which are utilized as in the real use as in the simulation in MuJoCo. The Kinova Gen3 comes only as the body of a manipulator, without the gripper. The gripper used is the 2F-85 by Robotiq [22]. The number 85 represents a 85 mm wide-open range. The specifications can be found in the Fig. 3.10. The important part of this gripper are its contact pads. Area of these contact pads is considered the only contact area

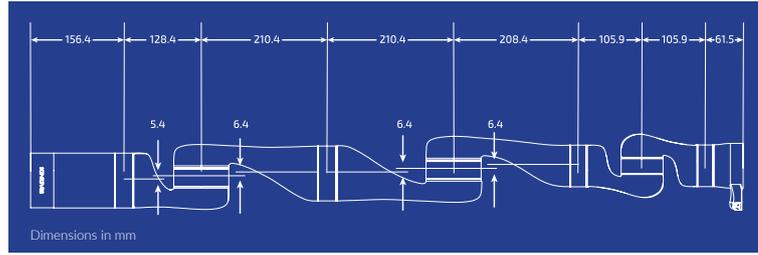


Figure 3.9: Blueprint describing relevant dimensions of the Kinova Gen3 7DOF manipulator [21].

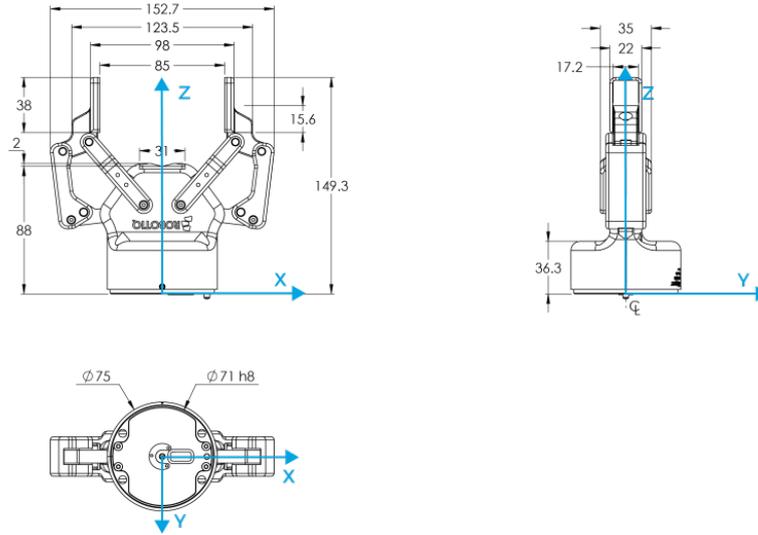


Figure 3.10: Detailed blueprint with all sizes dimensioned [22].

with the cubes. If the grasped object is too big, the gripper may squeeze the cube in between the hard parts of the gripper’s inner knuckles, which may distort the already noisy measurement. To avoid such cases, all of the objects are of the same dimensions: $56 \times 56 \times 56$ [mm]. For a more in depth insight into the reference data, including mass and measured reference values, see Tab. 3.2.

Material	Dens. [kg m^{-3}]		Elast. [kPa]		Side [mm]	Mass [g]
	μ_{ref}	σ_{ref}	μ_{ref}	σ_{ref}	a	m
Black Cube*	451.09	28.53	17.25	14.11	56	85
White Cube	237.88	33.04	15.35	7.25	56	39
Yellow Sponge	25.62	27.35	7.18	5.66	56	4

Table 3.2: Parameters of the real-life reference objects. * The Black Cube has an additional weight ring mounted on the bottom. The dimensions of the ring are neglected.

The objects were weighed and squeezed 10 times each, to extract an average mean and a standard sample deviation for the object properties, see Fig. 3.12.

The weighing process is done identically as in the simulation—as described in the Section 3.1.3. The objects were selected to create a setup that would make it easy for the two objects to be ambiguous in one property and distinguishable in the other. So two identical cubes, one White (Kinova White Cube) and one Black (Kinova Black Cube) were used. Their elastic properties are almost identical. To make those two objects measurably distinguishable, a metal ring was adhered to the bottom of the Black Cube. See Fig. 3.11(c). The Black Cube was measured for elasticity *with* the ring adhered, to account for any elastic property changes. It turned out that the ring seemingly made some structural changes to the way the cube is squeezed, probably pulling it more together, providing greater resistance against squeezing. This reflected on greater measurement inconsistency and a slightly larger Young’s modulus. A reference database based on these objects is created for the purposes of

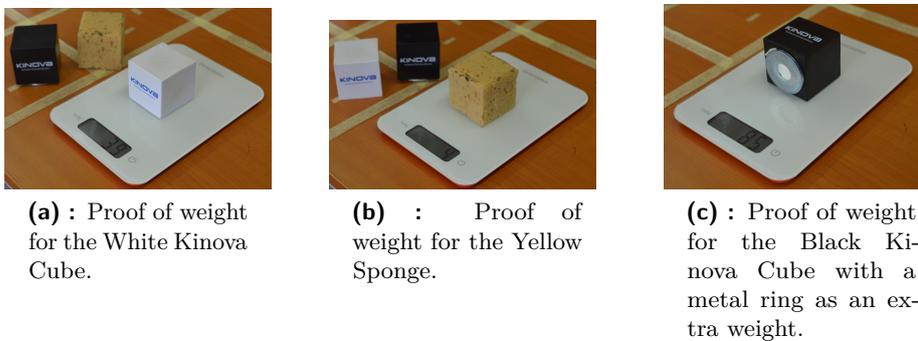
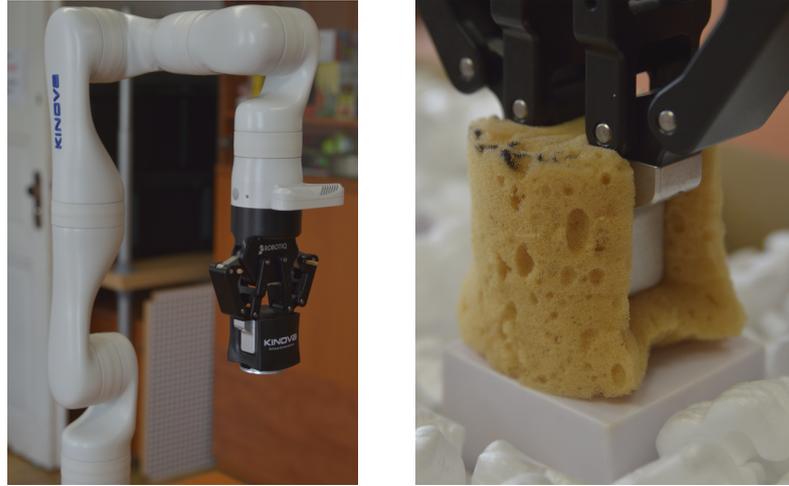


Figure 3.11: Three different cubes weighed, one with additional weight.

action selection. See, that on the Fig. 3.13 the Gaussians are cut-off at the y-axis. This is due to physical absurdity of some properties being negative. Although for the purposes of differential entropy computation, the whole property distribution needs to be offset to account for the distribution as a whole, not only a *visible* part, while handling the offset manually while computing the Bayesian update.

■ 3.2 Probabilistic reasoning

To reason about unknown properties of the outside world from the perspective of a robot manipulator, a need arises for a robust world representation. To represent the world in a useful manner, we need to account for the imperfections of the robot’s sensors in a practical way, allowing us to further infer information. In an attempt to quantify the imperfections of perception (e.g. computer vision, haptic sensing), probabilistic reasoning comes in handy. The main focus of selecting exploratory actions is via computing the values of expected information gain. In the context of real life, there is no time for taking great many measurements just to understand, what to explore next. A more incremental way of incorporating the new information into a world model is needed. There is not much space for real *frequentist* reasoning,



(a) : Action: Weighing – of the Kinova Black Cube with ring.

(b) : Action: Squeezing – of the Yellow Sponge.

Figure 3.12: Showcase of two actions, weighing (a) and squeezing (b).

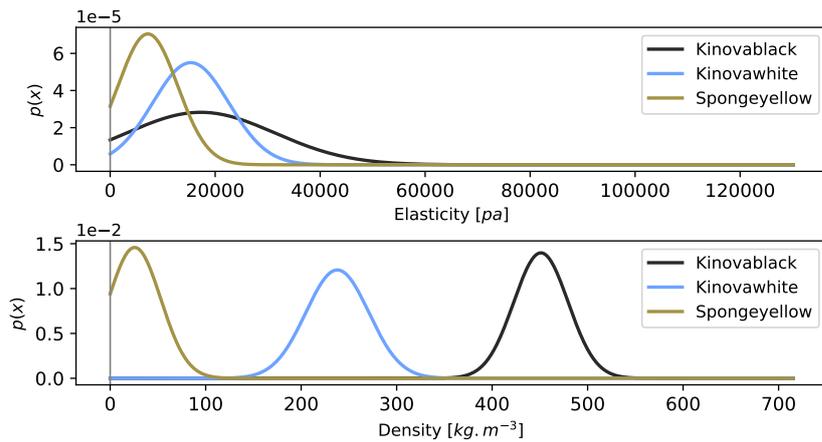


Figure 3.13: Reference of the real-world objects.

where we could perform N measurements and choose the one that is the most informative. The goal is to choose the measurement that is currently believed to be the most informative *in advance*. This set up calls for reasoning that also takes into account a belief, formally known as a *prior* [23]. The proposed formalism is called *Bayesian reasoning*.

3.2.1 Bayesian network

For a relevant and directed transfer of knowledge (information) between different domains, some kind of *information pipeline* is needed. With use of *Bayesian reasoning* an approach to solve this need is proposed: *The Bayesian*

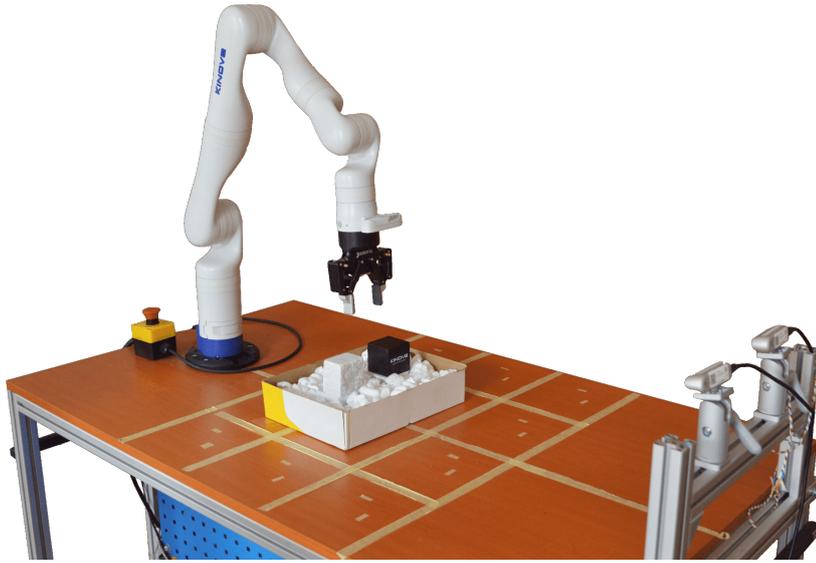


Figure 3.14: Photograph of the actual setup at the university laboratory.

network. As defined by S. Russel and P. Norvig [23, Ch. 14], the Bayesian network is a directed graph in which each node is annotated with quantitative probability information. In short, each node corresponds to a random variable; nodes are connected with arrows which show the directionality and each node has a conditional probability distribution that quantifies the effect of the parents on the node.

This definition is well-suited for representing random variables with well known probability distribution. Although our model also contains connections based on the laws of physics (Fig. 3.15a), where the relationships are certain. The *Object shape* node is not yet used; it is not in the scope of this thesis. Simplifying the network by omitting the *Object shape* node (as in Fig.3.15b) reduces the complexity significantly. Now, the volume is perceived to be independent and only being a multiplicative constant for the density approximation. With those simplifications, we may assume that with measuring weight we are actually measuring the material density scaled by some weight constant. Simply measuring the weight of an object without this simplification does not provide any meaningful information, if the object shape is not accounted for.

Regarding the terminology of stiffness versus elasticity, with our setup, only measurement of elasticity via Young's modulus is currently possible. Elasticity is the measure of *elastic* deformation, whereas stiffness is the measure of resistance to deformation regarding also the overall object shape. The Young's modulus describes the deformation without the effect of the physical shape (although it is inversely proportional to the cross-sectional contact area, see Eq. (3.1)) [15, Ch. 9].

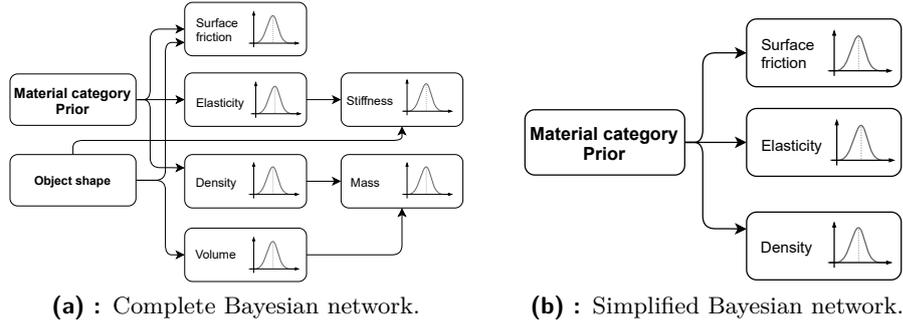


Figure 3.15: Simplification of the Bayesian network.

3.2.2 Bayes theorem

Exploring the object properties with *Bayesian reasoning* requires a formula for updating prior belief as the evidence accumulates. For this the *Bayes theorem* is used. It can be symbolically written as:

$$posterior \propto (prior \times likelihood) \quad (3.6)$$

Or more formally, following conventions by A.Gelman et al. in the Bayesian Data Analysis [24]:

$$p(\theta|y) \propto p(\theta)p(y|\theta) \quad (3.7)$$

where θ is the *examined property*, y is the *data*, $p(\theta)$ is the *prior distribution*, $p(y|\theta)$ is the *data distribution* and $p(\theta|y)$ is the *posterior density*, although still unnormalized. To gain posterior information, that is comparable with other (physical) domains, normalization is needed as follows:

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)} = \frac{p(\theta)p(y|\theta)}{\sum_{i \in I} p(\theta_i)p(y|\theta_i)} \quad (3.8)$$

where $p(y)$ is referred to as the *data* or *evidence*, or in the summation form, i represents one event (e.g. property or a category) from the set of I events (e.g. properties or categories). The evidence is the sum (or an integral in a continuous case) of all the possible events of θ weighted by each event's probability. Another way to express the nominator in the Eq. (3.8) is in form of *joint probability distribution*, which denotes how likely it is for two (or more) random variables to occur simultaneously, written as:

$$p(\theta, y) = p(\theta)p(y|\theta) \quad (3.9)$$

3.2.3 Bayesian inference

Inference is an act of creating a guess or forming an opinion based on information possessed beforehand. This is especially true for *Bayesian inference*. To infer in a Bayesian manner means to utilize Equation (3.8) in its discrete or continuous form according to a given prior and newly obtained information.

The equation (3.8) should be rewritten for the purposes of this example in the following way:

$$p(m|\hat{\varepsilon}) = \frac{p(m)p(\hat{\varepsilon}|m)}{\sum_{i \in I} p(m_i)p(\hat{\varepsilon}|m_i)} \quad (3.10)$$

where $p(m)$ is a short notation for $P(M = \text{material})$ and $\hat{\varepsilon}$ is the *noisy measurement* (and ε is the true value of the measurement). Using this equation to update our prior belief is called the *Bayesian update*. The probability of the noisy measurement given the queried material $p(\hat{\varepsilon}|m)$ can be obtained via marginalizing over the true measurement value ε as follows:

$$p(\hat{\varepsilon}|m) = \int p(\hat{\varepsilon}|\varepsilon)p(\varepsilon|m)d\varepsilon \quad (3.11)$$

where the $p(\hat{\varepsilon}|\varepsilon)$ is the probability density function of the noisy measurement and $p(\varepsilon|m)$ is the probability density function of the material's property, where ε is the distribution's mean value. Both of these variables are assumed to have normal distributions $\mathcal{N}(\mu, \sigma^2)$. The integral (3.11) can be solved either in closed form [25] or numerically. For better clarity, ε is exchanged for the μ_ε to further emphasize that the value in question is actually the mean:

$$\int p(\hat{\varepsilon}|\varepsilon)p(\varepsilon|m)d\varepsilon = p(\hat{\varepsilon}|\mu_\varepsilon)p(\mu_\varepsilon|m)d\mu_\varepsilon \quad (3.12)$$

$$= \int \mathcal{N}(\hat{\varepsilon}|\mu_\varepsilon, \sigma_\varepsilon^2)\mathcal{N}(\mu_\varepsilon|\mu_{\text{prior}}, \sigma_{\text{prior}}^2) \quad (3.13)$$

$$= \mathcal{N}(\hat{\varepsilon}|\mu_{\text{prior}}, \sigma_{\text{prior}}^2 + \sigma_\varepsilon^2) \quad (3.14)$$

This follows from the general properties of normal distributions and namely the fact that expected value (and variance) of a sum of two or more independent random variables is simply the sum of the expected values (or variances) of mentioned variables. Formally:

$$E[X_1 + X_2] = E[X_1] + E[X_2] \quad (3.15)$$

$$Var[X_1 + X_2] = Var[X_1] + Var[X_2] \quad (3.16)$$

In conclusion, to obtain the $p(\hat{\varepsilon}|m)$ the measurement $\hat{\varepsilon}$ is simply evaluated from the distribution as

$$\hat{\varepsilon} \sim \mathcal{N}(\mu_{\text{prior}}, \sigma_{\text{prior}}^2 + \sigma_\varepsilon^2) \quad (3.17)$$

.

3.2.4 Example of Bayesian inference

Using the methods described in the previous Section 3.2.3, an example can be created.

First, let us assume a categorical prior distribution described by a probability mass function (PMF). This function represents a discrete probability

distribution of the current belief, where the sum of the probabilities is always one. This belief represents a material category. As stated in sections before (namely 3.1.1), the variable space consists of four random variables representing the materials.

The prior PMF can, for example, take on such values:

$$\begin{aligned}
 P(M = \textit{Ceramic}) &= 0.7 \\
 P(M = \textit{Aluminium}) &= 0.15 \\
 P(M = \textit{Plastic}) &= 0.1 \\
 P(M = \textit{Steel}) &= 0.05
 \end{aligned} \tag{3.18}$$

A density measurement is performed, represented as a normal distribution:

$$\hat{\varepsilon} \sim \mathcal{N}_{\varepsilon}(\mu = 3150, \sigma^2 = 300) \quad [\text{kg m}^{-3}] \tag{3.19}$$

The categorical distribution represented by the PMF only represents the amount of belief among categories, although this information is not reflected into the overall shape (e.g. mean or variance) of the following distribution.

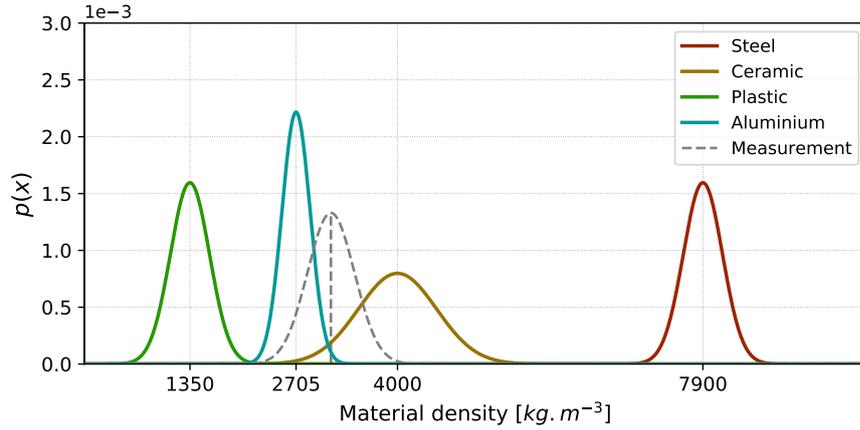


Figure 3.16: Material density – Prior distributions from reference values and a hypothetical measurement.

Using equations (3.12-3.14), versions of normal distributions (as seen on Fig. 3.16) are updated to take form (3.17), where the μ_{prior} (resp. σ_{prior}) is the mean density (resp. standard deviation) value for the material. This process is repeated for each material in the variable space. This makes the bell curves wider and shorter. That can be seen on figure 3.17. The distributions, now containing also the measurement error, are used. The measurement's mean is evaluated for each and every normal distribution in the variable space. On the figure, the function values are indicated by circles filled with color corresponding to the evaluated material.

The value of the measurement's mean in respect to each material m_i corresponds to the term $p(\hat{\varepsilon}|m_i)$ from the Eq. (3.10). So the *Bayesian update* can be, for example, shown for Ceramic:

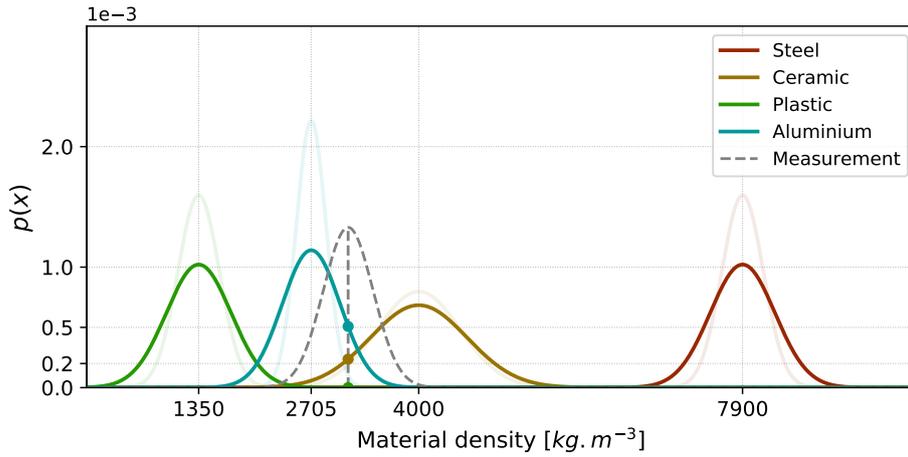


Figure 3.17: Widened distributions for each prior. Transparent Gaussians in the background indicate the previous shape of the distributions - before adding measurement error. The dots on the vertical line representing the measurement are the concrete likelihoods for each category with relation to the measurement.

$$P(M = \text{Ceramic}|\hat{\varepsilon}) = \frac{P(M = \text{Ceramic})P(\hat{\varepsilon}|M = \text{Ceramic})}{\sum_{i \in I} p(m_i)p(\hat{\varepsilon}|m_i)} \quad (3.20)$$

$$= \frac{p(c)p(\hat{\varepsilon}|c)}{p(c)p(\hat{\varepsilon}|c) + p(a)p(\hat{\varepsilon}|a) + p(p)p(\hat{\varepsilon}|p) + p(s)p(\hat{\varepsilon}|s)} \quad (3.21)$$

$$= \frac{0.7 \cdot 2.36 \cdot 10^{-4}}{0.7 \cdot 2.36 \cdot 10^{-4} + 0.15 \cdot 5.07 \cdot 10^{-4} + 0.1 \cdot 2 \cdot 10^{-8} + 0} \quad (3.22)$$

$$\doteq 0.6848 \quad (3.23)$$

Where:

$p(m)$ is short for $P(M = \text{Material})$

$p(\hat{\varepsilon}|m)$ is shot for $P(\hat{\varepsilon}|M = \text{Material})$

c , s , p and a are short for Ceramic, Steel, Plastic and Aluminium respectively

Repeating the process above for each material results in a complete step of *Bayesian update*. To show the updating ability of this framework, the exact same measurement and update are repeated ten times, resulting in ten steps of *Bayesian update*, shown in the Table 3.3. The relevance of the *Bayesian inference* starts to show just after a few first steps, flipping the belief around after two iterations. This shows, that no matter how imprecise the prior may be, the updating process is able to handle inconsistencies. However the amount of measurements needed to breach some confidence threshold is affected both by the prior consistency and measurement error. The greater the inconsistency and error, the more measurements are needed to converge the inference.

N-th step	Ceramic	Aluminium	Plastic	Steel
0	0.700	0.150	0.100	0.050
1	0.684	0.316	0.000	0.000
2	0.502	0.498	0.000	0.000
3	0.320	0.680	0.000	0.000
4	0.180	0.820	0.000	0.000
5	0.092	0.908	0.000	0.000
...
10	0.002	0.998	0.000	0.000

Table 3.3: 10 steps of the *Bayesian update*.

3.2.5 Obtaining a prior

To create a representation of current information, a distribution encompassing the reference data and the probability distribution of the material is desired. A one dimensional Gaussian Mixture Model (GMM) is constructed. All of the information about the distribution is kept upon construction and used for the purposes of Bayesian inference, as shown in the previous Subsection 3.2.4. Reference data are provided at the initialization—this data is not learned, nor updated in any way throughout the algorithm. For better clarity, see the provided visualization 3.18.

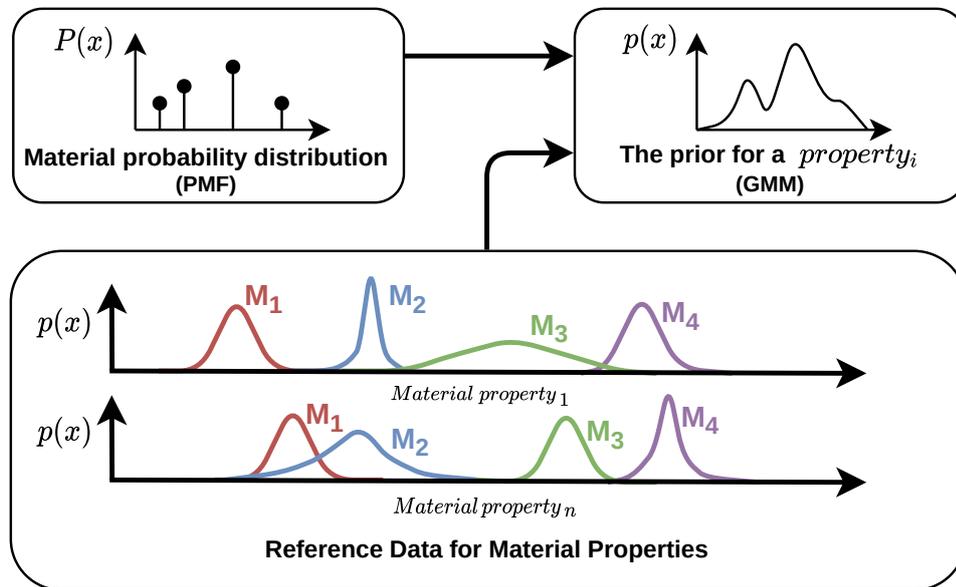


Figure 3.18: Flowchart of the prior Gaussian Mixture creation.

Following the diagram 3.18, two Gaussian Mixture Models, one for each property, are created. Both can be seen on the Fig. 3.19.

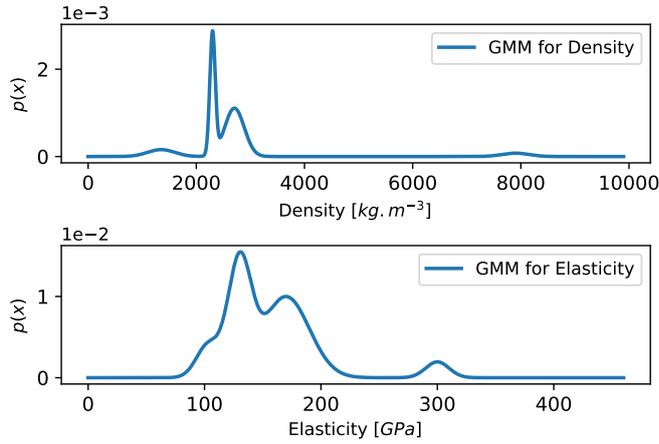


Figure 3.19: Example of a resulting Gaussian Mixture Model.

3.3 Action selection

Action selection is the core of this work. The action repertoire is shown in the Section 3.1.1. The goal is to purposefully choose an action from the repertoire based on the information known beforehand (the current prior distribution) and an expected value of measurement constructed via sampling from the *a priori* distribution. A posteriori distribution is used, when measurements have already taken place. It is important to note that the action selection is cyclic, so what is the *a posteriori* (or simply posterior) distribution in the i -th iteration, is perceived as the *a priori* (or simply prior) distribution in the following $i + 1$ -th iteration.

3.3.1 Information theory

To study and quantify the amount of learning that can be accomplished, tools and techniques described in the topic of Information theory can be utilized. For any action selection to take place, a decision based on some quantity must be made. Information theory studies the quantification, transfer, storage and general communication of the information. Information theory is built on top of a publication by Claude E. Shannon, A Mathematical Theory of Communication [26]. Information theory has practical implications in fields of statistical physics, computer science, probability and statistics and much more. T. M. Cover and J. A. Thomas describe Information theory as a formal representation of the extreme points of the set of all possible communication schemes [27, Ch. 1]. One extreme (minimum) is the Data compression limit, known as *entropy*, and the other extreme (maximum) is the Data transmission limit, known as *channel capacity*. The former is crucial for action selection, and will be discussed in the following sections.

3.3.2 Entropy and differential entropy

Entropy is a function of some probability distribution which underlies the communication form of the information. In simpler terms, the entropy, in the discrete case, represents the average amount of information needed to represent an event drawn from a probability distribution for a random variable. This can be viewed as the measure of “surprise”. The entropy of a random variable X is defined as

$$H(X) = - \sum_x p(x) \log_2 p(x), \quad (3.24)$$

where $p(x)$ stands for the probability mass function. The expression (3.24) is called *discrete information entropy*, or shortly *entropy*. The unit of the discrete information entropy is *bit*. It is derived from the base of the logarithm (e.g. when a natural logarithm is used, the unit is called a *nat*). Alternative notation may be used, where the properties of logarithm are used, $H(X) = \sum_x p(x) \log_2 \frac{1}{p(x)}$.

The Eq. (3.24) can be also written for the continuous form as

$$H(X) = - \int_x f_p(x) \log_2 f_p(x), \quad (3.25)$$

where the $f_p(x)$ stands for the probability density function. The expression (3.25) is called *differential information entropy*, or shortly *differential entropy*. The differential entropy is uniteless. Although similar looking, information entropy for discrete and continuous cases are not interchangeable.

David J.C. MacKay in his book *Information Theory, Inference and Learning algorithms* proposes usage of a term *ensemble*, instead of a sole term *random variable*, [28, Ch. 2]. The author defines an ensemble X as a triple $(x, \mathcal{A}_X, \mathcal{P}_X)$, where the outcome x is the value of a random variable, which takes on one of a set of possible values, $\mathcal{A}_X = \{a_1, a_2, \dots, a_I\}$, having probabilities $\mathcal{P}_X = \{p_1, p_2, \dots, p_n\}$, with $P(x = a_i) = p_i, p_i \geq 0$ and $\sum_{a_i \in \mathcal{A}_X} P(x = a_i) = 1$.

This notation is best suited for a discrete random variable or a set of random variables. In this work, discrete entropy and differential entropy are both used, therefore a uniform and brief notation \sum_x (or \int_x) is used (as opposed to the notation from the aforementioned book $\sum_{x \in \mathcal{A}_X}$). The fact that a random variable takes on values with some non-negative probability where the sum of probabilities is equal to one is generally assumed in both continuous and discrete cases.

3.3.3 Action selection algorithm

For a robot to autonomously learn object properties through exploration, an algorithmic approach based on some metric is needed. As outlined earlier, the metric for the action selection in this work is, logically, the expected change of entropy, i.e. the information gain. Randomly choosing an action is also an option, but the information gain of a repeated measurement (assuming the

consistency of measurement error) is often times close to zero. In this way, randomly choosing an action despite of having the information about measurement history would not be effective, although vastly easier to implement. To decide between specific actions (i.e. what property to measure), various approaches are possible.

3.3.4 Differential entropy of a property belief

Computing the *differential entropy of a property belief* is done in a continuous way by numerically approximating the equation (3.25) with its discrete form (3.24). To compute this entropy a probability density function (PDF), which would represent the current belief, is needed. This PDF is hard to obtain in a closed form with such Gaussians, as seen on Fig. 3.16. A numerical approach was chosen, where the PDF as seen in Fig. 3.20b, is constructed as a sum of each $\mathcal{N}(\mu_i, \sigma_i^2)$ weighted by its probability. This results in a distribution called Gaussian Mixture Model, which contains information about the chosen material property, their mean values μ_i , their variances σ_i^2 and also the probability for each material being the correct category for the explored object. For better clarity see Fig. 3.18. This material category probability is recorded in the probability mass function (PMF), as seen on Fig. 3.20(a). On the same picture, a material referred to as a *joker* is mentioned. This is a supplemental material category, that represents any other unknown material in the prior. More in-depth description and explanation is provided in the following sections.

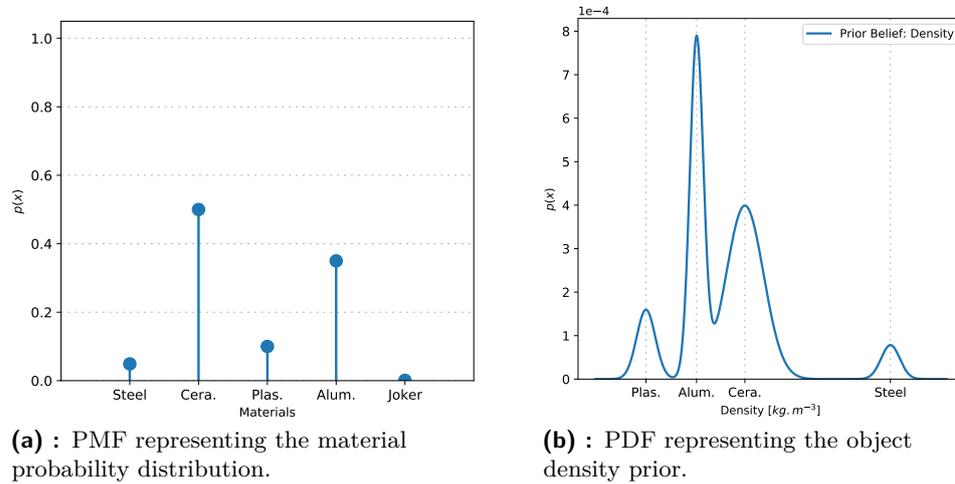


Figure 3.20: Comparison of the material probability distribution (a PMF) and the object density prior (a PDF).

To clarify the figure above, it is obvious from (a) that the material *Ceramic* has the highest probability of being the correct material category for the explored object. However, the maximum peak in (b) is above *Aluminium*. This is simply because of the fact that the *Ceramic* has a greater variance in

parameters (in the sense of how much can material properties change, while still being categorized as *Ceramic*), opposed to the *Aluminium*, which is very strict in its material properties definition, and therefore has smaller variance.

The differential entropy of a property belief can be formally written as $H(\mathcal{P})$, where the \mathcal{P} stands for “property”. As mentioned before in the Section 3.3.2, differential entropy has no units as opposed to discrete entropy. The approximation of the integral with a sum does not mean that the entropy is now described in bits. The entropy is computed as

$$H(\mathcal{P}) = - \sum_x h \cdot \bar{f}_{\mathcal{P}}(x) \log_2 \bar{f}_{\mathcal{P}}(x), \quad (3.26)$$

where the h is the step size (or bin size), and the $\bar{f}_{\mathcal{P}}(x)$ is the numerical approximation of the continuous probability density function $f_{\mathcal{P}}(x)$.

3.3.5 Material entropy

To actually make an informed decision based on the information gain, an educated estimate is needed to update the prior distribution with Bayesian update. The educated estimate is obtained by simulating measurements. This is called measurement emulation and is done simply by sampling the prior distribution expanded with the measurement error. This is done in closed form, similarly as described in the Subsection 3.2.5, where each single Gaussian of the Gaussian mixture is widened as in the Eq. (3.17). Sampling the distribution draws N points, that follow the probability density function. In other words, the higher the peak, the more likely is the point to be drawn. This can be seen in the Fig. 3.21. For purposes of this work, let $N = 100$.

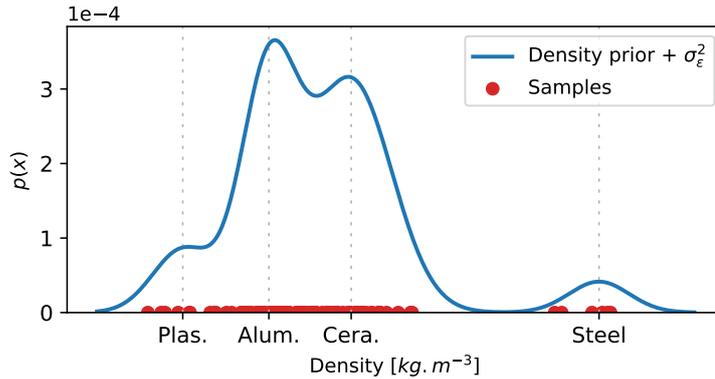


Figure 3.21: Sampling from the emulation.

The red dots in the Fig. 3.21 represent an expected measurement. In the sections before, a notation $\hat{\epsilon}$ for the *noisy measurement* value was used. Now $\hat{\epsilon}$ will be used to denote the *emulation of the noisy measurement* value. The emulation represents an expected measurement based on the prior knowledge. That is the reason why the prior PDF with incorporated measurement error

is used for the measurement sampling. It is worth noting that the actual measurement is expected to be a single normal distribution. The reason the samples in the figure above follow a Gaussian mixture distribution and not a single Gaussian is that the confidence (i.e. the material belief represented with the PMF) also plays role in the shape of the PDF. Simply put, we believe the measurement could measure various materials, and we expect to be so with some probability for each material.

Computing the *Material entropy* is not difficult. The probability of each material is expressed in the probability mass function, and computing its entropy means only to use the Equation (3.24) for discrete probability distributions. But to learn some information, (i.e. compute the information gain) an *Average expected entropy* is needed. The information gain is computed as

$$H(M) - E[H(M|\hat{\mathcal{P}})], \quad (3.27)$$

where $H(M)$ is the Material entropy and $E[H(M|\hat{\mathcal{P}})]$ is the Average expected entropy. The M stands for “material”, \mathcal{P} stands for “property” and $\hat{\mathcal{P}}$ stands for the property emulation. In Fig. 3.15 we have previously shown that the material properties \mathcal{P}_i are the children of the PMF, not the other way around. This means that our current material belief dictates the expected properties, and the expected properties dictate the parameters of the PMF only through the Bayesian update.

To obtain the Information gain from the Equation 3.27, we first need to compute the Average Expected Entropy value as

$$E[H(M|\hat{\mathcal{P}})] = \frac{1}{N} \sum_n^N H(M|\hat{\epsilon}_n) \quad (3.28)$$

where the $\hat{\epsilon}$ represents individual measurement emulations for the current property \mathcal{P} , M stands for the material, N is the number of samples and $\hat{\mathcal{P}}$ is the emulated property.

To compute each entropy emulation of a property for each material $H(M|\hat{\epsilon}_n)$ the Bayesian update must be done first as described in the Subsection 3.2.3. Afterwards the entropy is computed according to the Equation (3.24). After this is done for all of the samples, an average of those possible entropy outcomes is made. So in result, the Information gain does not represent how much we *will learn* or *have learned*, but how much do we *expect to learn on average* if a measurement is taken in the domain of this property. The step from the Probability Mass Function to the Probability Density Function was previously explained in the Subsection 3.2.5. The general outline of the whole process is shown in the Figure 3.22 below.

■ 3.3.6 Differential entropy of a property with relation to a property emulation

Computing the property entropy provides yet another way to assess the amount of gained information. As seen in the Subsection 3.3.4, it is also

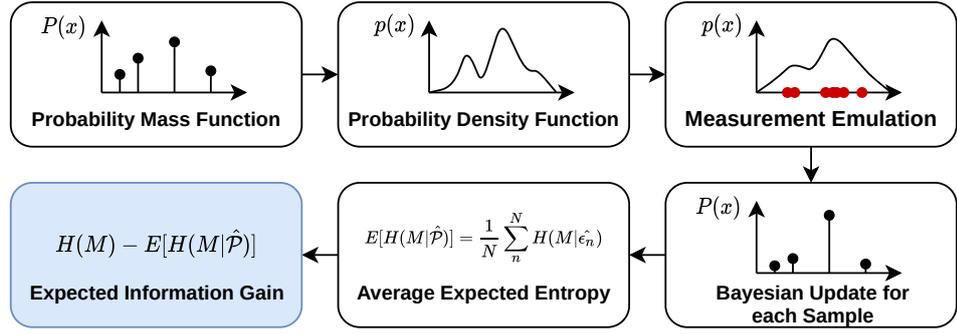


Figure 3.22: Object material category – outline of expected information gain calculation.

possible to compute the entropy for a continuous variable. The whole property belief is a Gaussian mixture, which is the continuous distribution referred to. Computing solely the differential entropy does provide some information, but the information is not alone sufficient. It is impossible to describe a continuous variable with finite number of bins, because there will always exist a finer binning describing the variable. The difference between the current differential entropy and the *expected differential entropy* is what matters in this case. This approach is similar to the Subsection 3.3.5, just in a continuous sense. The *Expected differential information gain* is the difference between the prior differential entropy and the expected differential entropy, formally written as

$$H(\mathcal{P}) - E[H(\hat{\mathcal{P}})], \quad (3.29)$$

where \mathcal{P} is the property and $\hat{\mathcal{P}}$ is its emulation. The emulation, as opposed to the discrete measurement emulation, now encompasses all possible expected measurements at once, in a single distribution.

Both $H(\mathcal{P})$ and $H(\hat{\mathcal{P}})$ are calculated as stated in the subsection about Differential entropy of a property belief 3.3.4. The expected differential entropy $E[H(\hat{\mathcal{P}})]$ could be obtained as in the previous subsection, although it is possible to account for all of the possible expected measurements by using the whole emulation without actually sampling from it. This makes the emulation more robust and no average of measurements is needed. Although, this approach brings an important caveat.

Once the PDF is created and updated with an actual, real-world measurement, the comfort of closed-form solutions is lost from that point. The real-world measurement is incorporated via element-wise multiplication and then normalized. This approach is nonetheless valid, but makes it very difficult to create an emulation (i.e. prior widened with the measurement error) in a closed form, as it was in the previous Subsection 3.3.5. Instead, a convolution of the prior distribution with a Gaussian representing a general measurement is done. The general measurement is created as a Gaussian with the mean in the middle of the *working window* and the sigma of the measurement – the measurement error. The *working window* refers to the numerical range of

bins, upon which the Gaussian Mixture Model (GMM) is constructed. In python's `numpy` library, this can be represented as: `numpy.linspace(start, end, num_of_bins)`. A window of the same length is used for both GMM and the general measurement. When the GMM and the general measurement are constructed, they are convolved resulting into a distribution referred to, in the previous section, as the *emulation*. More formally written as

$$\bar{f}_\epsilon(x) = \bar{f}_\mathcal{P}(x) * \bar{g}_\epsilon(x), \quad (3.30)$$

where $\bar{f}_\mathcal{P}(x)$ is the Gaussian Mixture Model for the property \mathcal{P} , $\bar{g}_\epsilon(x)$ is the general measurement distribution and $\bar{f}_\epsilon(x)$ represents the *emulation* distribution. For the sake of notation consistency, bars were added above all of the distributions, to address the discrete implementation of those operations.

Calculating the *posterior* distribution for further differential entropy calculation is done by element-wise multiplication of the *prior* and the *emulation*:

$$\bar{y}(x) = \bar{f}_\mathcal{P}(x) \odot \bar{f}_\epsilon(x). \quad (3.31)$$

After the *posterior* is acquired, the entropy $E[H(\hat{\mathcal{P}})]$ is computed using Eq. (3.26). The rough outline of this process is shown in the Figure 3.23.

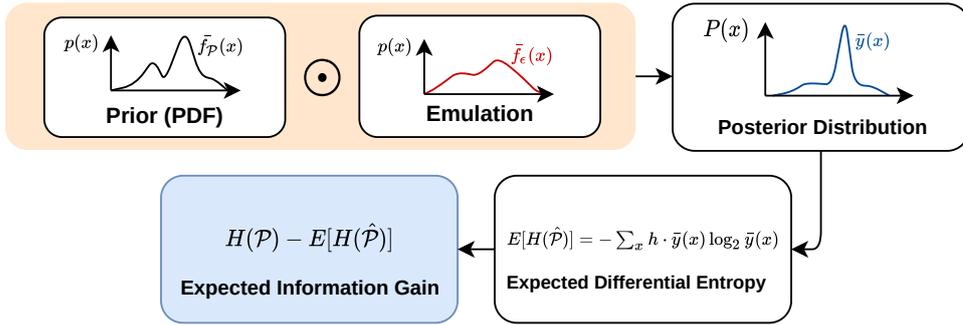


Figure 3.23: Outline of the Expected differential information gain calculation.

3.3.7 Bayesian Message Passing

It is important to clarify the current chain of events in the Bayesian network. Currently the Bayesian network is only updated back to the Material Category Prior node. This provides new information for the *category only*. The information from one child node (e.g. Density) which got to the parent node (e.g. Material Category) does not travel further into other child nodes. Such process is called *Belief propagation* and requires a *message-passing* algorithm. This would be very beneficial for the robustness of the belief propagation. The current belief is propagated only in the parent-child pair in each measurement episode. There is currently no message-passing in between the child nodes. This is left for further work, as the difficulty level of such algorithm is out of the scope of this thesis. The belief propagation can be seen on the Fig. 3.24.

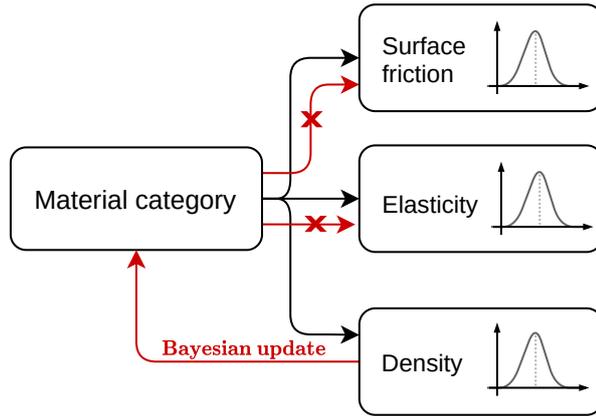


Figure 3.24: Network without the message passing in between child nodes.

A proper visualization of the inference process can be seen on the Fig. 3.25. In the left top corner, there is a red arrow pointing from the Initial PMF: Π_0 to the Initial PDFs $\rho_{1_0} \cdots \rho_{n_0}$. This symbolizes the creation of the initial PDFs from the initial prior PMF, as described in the Subsection *Obtaining a prior* (3.2.5). The first decision of action selection algorithm was, for example, to measure *property*₁. This measurement updates the Material Category prior (recall the previous Fig. 3.24) and the appropriate property PDF, in this case ρ_1 . Other PDFs ρ_n are not changed. They would only change, if aforementioned message-passing algorithm would be implemented. So those PDFs are in a sense detached from the Material category, they only govern the action selection Modes (Modes will be described later) and can provide argmax of the latest belief.

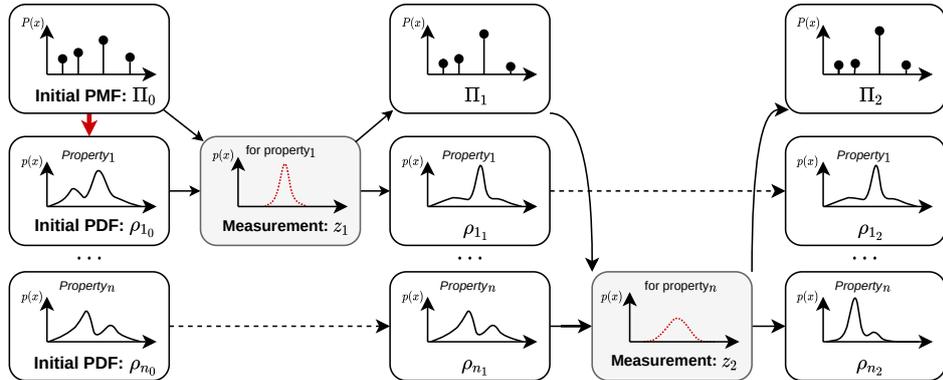


Figure 3.25: Visualization of 2 exemplary measurement episodes.

3.3.8 The Action selection algorithm

To select the most suitable action with the highest information gain, a few different metrics can be used. Two similar algorithms that differ in whether they seek to maximize the information gain of discrete or continuous

probability distributions are presented below.

■ Mode 1: Material category optimization

In the Algorithm 1, only the Discrete Information Gain is optimized for. This results in a discrimination process that seeks changes to the probabilities of the material category (the PMF) and ignores the uncertainty of the continuous probability distributions for individual object properties. In other words, this approach acknowledges that some measurement is expected on average to make greater differences between the probabilities of each material. Alg. 1 utilizes theory from the Section 3.3.5 for computing the discrete entropy, referred to as `ComputeDiscreteEntropy()`. The `BayesianUpdate()` is applied in the form described in the Subsections 3.2.3 and 3.2.4.

Algorithm 1: Discrete Material Information Gain optimization.

```

1 max_info_gain := 0
2 if prior is available then
3   | PMF := prior
4 else
5   | PMF := uniform_prior
6 end
7 assert ground_dict exists
8 foreach property in ground_dict do
9   | GMM := BuildGaussianMixtureModel(property, ground_dict, PMF)
10  | prior_disc_entropy := ComputeDiscreteEntropy(PMF)
11  | GMM $\hat{\sigma}$  := ApplyMeasurementError(GMM)
12  | samples := Sampler(GMM $\hat{\sigma}$ )
13  | foreach sample in samples do
14    | temp_PMF := BayesianUpdate(PMF)
15    | disc_entropy := ComputeDiscreteEntropy(temp_PMF)
16    | append disc_entropy to disc_entropies_list
17  | end
18  |  $E[\textit{disc\_entropy}]$  := Mean(disc_entropies_list)
19  | info_gain_disc := prior_disc_entropy -  $E[\textit{disc\_entropy}]$ 
20  | if info_gain_disc > max_info_gain then
21    | max_info_gain := info_gain_disc
22    | planned_measurement := property
23  | end
24 end
25 return planned_measurement

```

■ Mode 2: Property optimization

Optimizing for the Differential Property Information Gain is another metric to gauge the expected information gain. The differential information gain ‘lives’ in a different variable space as opposed to its discrete counterpart. One of the reasons is the fact that the differential entropy may be negative [27, Ch. 8]. That stems from the basic definition of entropy. Discrete entropy

can describe only a discrete phenomenon in the discrete world, in a finite number of steps, i.e. bits (or nats). To describe the information in the same manner as the discrete entropy does, means to slice the continuum into a discrete world with many bins. This is called quantization. And the measure of differential entropy is only as good as its quantization. This means that no matter how fine your quantized bins are, you can *always* go finer. this implies that with finer bin resolution comes more precise approximation of the differential entropy.

Algorithm 2: Differential Property Information Gain optimization.

```

1  $max\_info\_gain := 0$ 
2 if prior is available then
3   |  $PMF := prior$ 
4 else
5   |  $PMF := uniform\_prior$ 
6 end
7 assert ground_dict exists
8 foreach property in ground_dict do
9   |  $\bar{f}_{\mathcal{P}} := \text{BuildGaussianMixtureModel}(property, ground\_dict, PMF)$ 
10  |  $\bar{f}_{\epsilon} := \text{ApplyMeasurementErrorToGMM}(\bar{f}_{\mathcal{P}})$ 
11  |  $\bar{y} := \text{ElementWiseMultip}(\bar{f}_{\mathcal{P}}, \bar{f}_{\epsilon})$ 
12  |  $prior\_diff\_entropy := \text{ComputeDifferentialEntropy}(\bar{f}_{\mathcal{P}})$ 
13  |  $posterior\_diff\_entropy := \text{ComputeDifferentialEntropy}(\bar{y})$ 
14  |  $info\_gain\_diff := prior\_diff\_entropy - posterior\_diff\_entropy$ 
15  | if  $info\_gain\_diff > max\_info\_gain$  then
16  |   |  $max\_info\_gain = planned\_measurement$ 
17  |   |  $planned\_measurement := property$ 
18  | end
19 end
20 return  $planned\_measurement$ 

```

The Algorithm 2 describes the optimization for the differential information gain, which is calculated for each currently measurable property. This algorithm simply describes the whole process from the Subsection 3.3.6. The notation in the algorithm is consistent with the notation in the aforementioned subsection.

■ Mode 3: Hybrid sum optimization

This type of optimization does not need its own algorithmic description. It simply optimizes for the sum of the Discrete Material and Differential Property Information gain. This is not mathematically correct, as the discrete and differential information gains live in different variable spaces (see the previous part). However, it is possible to use this unitless number as an *abstract proxy metric* for the optimization. This would mean that the choice of the next planned action would be judged based on the sum of $info_gain_disc$ and $info_gain_diff$ from the Algorithm 1 and 2 respectively. The property with the highest expected hybrid sum is chosen.

■ 3.3.9 Measurement uncertainty and a placeholder category

The main goal of the action selection framework is to gain more information—or reduce the uncertainty—about an object being inspected. At the beginning of this work, it was stated that the variable space will be consisting only of four materials. That is correct, although there may come a situation where an object is measured, that does not fit with any of the referential data. The algorithm would slowly converge to the closest variable in the variable space, although that may not be appropriate. For such special cases, a placeholder category is introduced.

■ Unexpected Real Measurement

It is important to expect the unexpected. The Bayesian update has also some shortcomings. The denominator of the Bayes theorem (see Eq. (3.10) and Eq. (3.8)) is a discrete sum, which assumes that all of the categories are known and defined. A problem occurs, when a measurement of a material that shows much greater or smaller quantity of property than currently known, is measured. For example, let us assume, that the measurement error is infinitely small, effectively rendering the measurement absolutely trustworthy. A measurement of the elasticity of some diamond gemstone is taken, returning the Young's modulus of approximately 1000 GPa. The hardest material that we currently have in the reference database (and consequently in the PMF) has the Young's modulus of for example 300 Gpa. Such "diamond measurement" would be numerically unstable (as the inferred value would be very close to zero).

Even if it was possible to make the computation numerically stable, the Bayesian update would just update the probability of the "hardest" material (i.e. a material closest to the current measurement) and would therefore loose precision. After a few steps of repeated measurements and inference, the probability in the PMF would converge to the category closest to the measurement. This may be perceived as a feature, not a problem, because the algorithm would come as close as possible to describing the real object measured, **in terms of the current property**.

This naïve approach updates the probability of a material, but the interdependency between material properties does not need to be straightforward. For example, an object, that is very stiff does not need to be heavy (e.g. carbon fibre) and vice versa. So the verdict of the algorithm should not be the material closest to the measurement, but an unknown material. This unknown "placeholder" material will be referred to as a *joker* for apparent reasons.

The joker is a uniform distribution that helps to avoid numerical problems and allows the probability to converge to an unknown category. In the case, where the measurement mostly resembles the unknown category throughout the properties, the information about the mean value in each property and the attributable sigma for the property would get lost. For such cases a

distribution, where the measurement is incorporated directly numerically (not indirectly through the discrete form of Bayesian update) is kept. This distribution is kept for each property separately, recall Fig. 3.25. The visual representation of this uniform *joker* distribution can be seen on the Fig. 3.26.

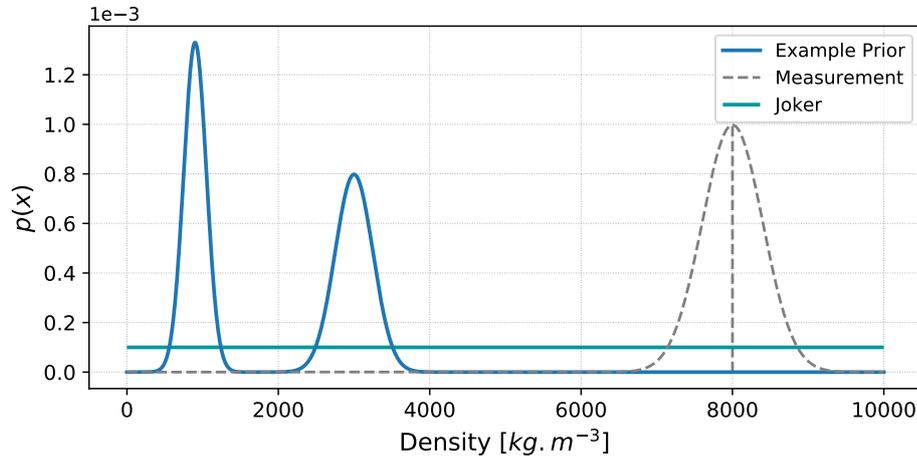
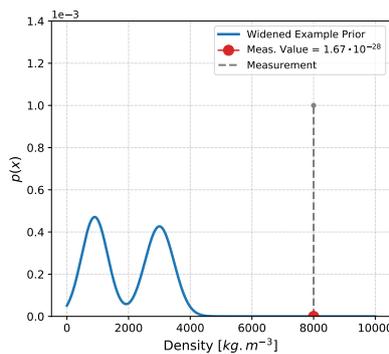
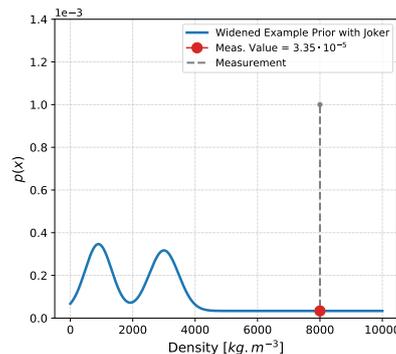


Figure 3.26: An example showing some arbitrary prior distribution, measurement distribution and a uniform distribution labeled *joker*.

In the Fig. 3.26, the measurement uncertainty σ is not accounted for in the *Example Prior* distribution. The Gaussians are shown as is. To show the numerical shortcomings of a possible object measurement of an unknown category, the Subfigures (a) and (b) in the Figure 3.27 were created.



(a) : The widened example prior (including the measurement error) without the *joker* uniform distribution yields a value very close to zero, that is $p(x) = 1.67 \cdot 10^{-28}$.



(b) : The widened example prior (including the measurement error) with the *joker* uniform distribution yields a significantly more reasonable number, that is $p(x) = 3.35 \cdot 10^{-5}$.

Figure 3.27: Comparison of some example prior distributions with and without the *joker* uniform distribution added.

3.3.10 Connection to IPALM database

In the IPALM project [3], various objects from various datasets are used for the project related experiments. The most populated dataset is the YCB dataset [29], which was also used to teach a neural network detectron 2 for the prior acquisition [30, 31]. The YCB dataset consist of everyday objects such as plates, cutlery, a water pitcher, a hammer, a tennis ball and many more. Generating good gripper poses for grasping such objects is difficult and still in progress by other members of the IPALM project. Simply put, those objects are too complex and therefore too difficult to measure in this thesis. More simple objects such as cubes are used to rule out difficulties with shape and non-homogenous density (e.g. a plastic bottle filled with air). For future use, a common structure for measurements, model properties and database reference was co-developed with other IPALM members. JavaScript Object Notation (JSON) files are used for cross-platform data storage. The goal for common data structures is unification and flexibility. If common ways of storing data are implemented, the parsing of such file is then straightforward. An example of such unified data structure can be seen in the Listing 2.

```
[{
  "object_name": "item",
  "measurements": [
    {
      "name": "density",
      "value": {
        "sigma": 1000,
        "mean": 7337.799079987667,
        "units": "kg.^{-3}"
      }
    },
    {
      "name": "elasticity",
      "value": {
        "sigma": 1000,
        "mean": 4969.761122576146,
        "units": "Pa"
      }
    }
  ],
  "object_id": "1"
}]
```

Listing 2: Example of a measurement data structure.

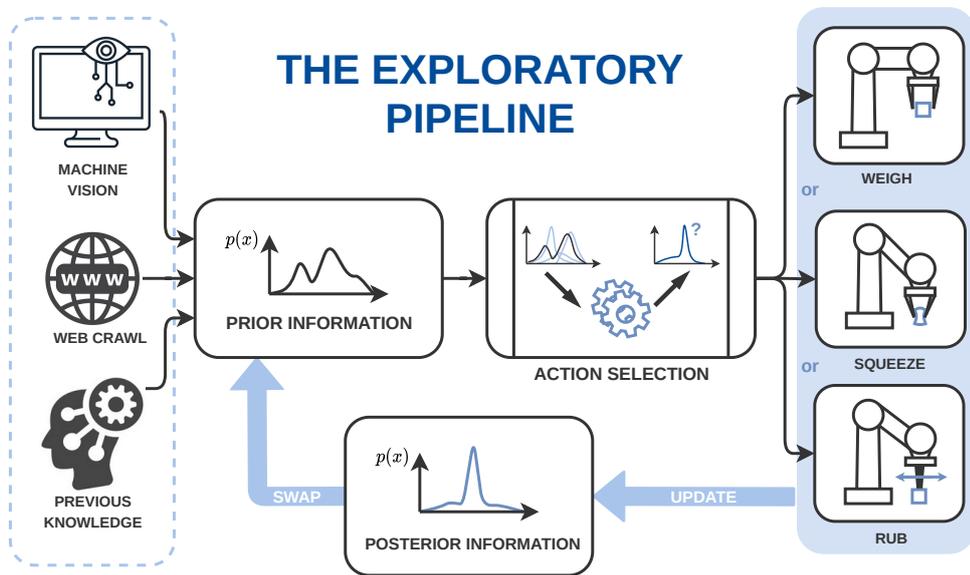


Figure 3.28: An overview diagram of the action selection pipeline.

Chapter 4

Experiments and Results

The action selection governs what exploratory actions are chosen to learn about properties of an object and possibly disambiguate between priors—it does not provide classification directly. The updating of the belief is done with the Bayesian update from a provided measurement. The belief (previously described in the form of the Gaussian Mixture Model or a Probability mass function describing the probabilities of known materials) is used after an arbitrary number of measurements to perform classification. The convergence of the classification is used as a proxy metric to judge the performance of the action selection.

The algorithm has undergone various tests in different circumstances. Firstly, a fully virtual experiment was conducted. This type of experiment tests the core of the algorithm, while having manual control over the simulated measurements. The simulated measurement mean was perturbed with random uniform noise, which substituted for the wider range of items described in some more vaguely defined categories, such as ceramic or plastic. That decision was made, because various items of slightly different properties are grouped and called ceramic, although it might be actually made of silica, resin, porcelain, various alloys and so on. All of the measurements were constructed manually, perturbed with aforementioned noise and the measurement sigma was chosen according to criteria described in the Section 4.1. The perturbation is added to the measurement mean. The density perturbation is randomly chosen from the range of $\langle -20, 20 \rangle$ bins (where the length of the density window is ≈ 9000 bins) and the noise for elasticity is chosen from a smaller range of $\langle -10, 10 \rangle$ bins (where the length of the elasticity window is ≈ 500 bins, but the material definitions for elasticity are much wider, which is noticeable on the Fig. 3.19 from the introductory Section 3.2.5).

Secondly, the other variation of testing was done also virtually, but with the help of a modern real-time physics simulator. The nature of the simulation brings its own perturbation, which left great opportunity for the action selection demonstration. For more details on the implementation, see the section on Virtual setup 3.1.2. The nature of the perturbation is elaborated on later.

For evaluation purposes a terminology was established as follows:

- **Correct prior** - a prior is perceived as correct, when the most likely category is the same as the *Truth*
- **Incorrect prior** - a prior is perceived as incorrect, when the most likely category differs from the *Truth*.
- **Proper measurement** - a measurement is considered proper, if the measurement error is in range: $\sigma_\varepsilon \in (\sigma_{\text{ref}}, 2\sigma_{\text{ref}})$.
- **Noisy measurement** - a measurement is perceived as noisy, if the measurement error is larger than $3\sigma_{\text{ref}}$.
- **Convergence** - if the algorithm ends with the most likely category, where the $P(M) \geq 0.6$ and the category corresponds with the *Truth*, it is perceived to converge to the correct category.
- **Divergence** - if the algorithm ends with the most likely category that differs from the *Truth* with $P(M) \geq 0.6$, or if the $\text{argmax}(P(m)) = \text{joker}$, it is perceived to diverge from the correct category.
- **Weak convergence** - if the algorithm ends with the most likely category, that corresponds with the *Truth* and $P(M) < 0.6$, it is perceived to converge weakly to the correct category.

The terminology above is then completed with a uniform prior and six different contexts of measurement type an prior type are created as follows:

1. Correct prior, proper measurement
2. Incorrect prior, proper measurement
3. Uniform prior, proper measurement
4. Correct prior, noisy measurement
5. Incorrect prior, noisy measurement
6. Uniform prior, noisy measurement

4.1 Virtual Experiments

To showcase the workings of algorithms from the Section 3.3.8, we conduct an exemplary experiment. Experiments are run in aforementioned varied settings—contexts, to measure the performance in terms of a correct classification rate, although the algorithm is not inherently a classifier. Even with a wrong classification, the algorithm might learn valuable information, for example, in the form of the argmax of each property belief, which yields the most likely mean value. For each context and each mode, 10 steps of the algorithm were done. Each test was repeated 5 times to rule out random flukes. In total, 90 tests are provided (6 contexts \times 3 modes \times 5 repetitions = 90).

*Note: It is necessary to point out, that the criteria for **Proper** and **Noisy measurement** are only usable, if the examined object has known properties.*

Otherwise, no assumption of proper or noisy measurement can be made, if there exists no clue, what to compare the deviation with, in the reference bank.

■ Correct prior, proper measurement

Such a setting does not provide any great challenge, since everything is set up right. In this scenario all three Optimization Modes (Mode 1, 2 and 3 from the Subsection 3.3.8) converged to the correct category.

■ Incorrect prior, proper measurement

This setting is actually very similar to the *Correct prior, proper measurement*, because the Bayesian update will overcome the wrong prior. Although the Mode 1 diverged in 3/5 tests, Modes 2 and 3 converged every time.

■ Uniform prior, proper measurement

Uniform prior is a very useful test case, as it really narrows down into what is the entropy selection worth. Mode 3 converged only in 1/5 tests, Mode 1 did not converge at all, but Mode 2 converged in all 5 tests.

■ Correct prior, noisy measurement

This setting was no problem for the Modes 2 and 3, although the strictly discrete Mode 1 did diverge in 5 out of 5 times.

■ Incorrect prior, noisy measurement

Incorrect prior and noisy measurement are quite a difficult setting, as there is not much to work with. Modes 1 and 3 diverged completely each time, Mode 2 is on the edge between weak convergence and convergence in 5/5 tests.

■ Uniform prior, noisy measurement

In this setting, only Mode 2 was successful, with 5/5 converged tests. Mode 1 and 3 diverged.

■ Overview in tables

In the following tables, the results of the 90 tests are provided. The detailed logs from the testing are available in the online GitLab repository [10]. Letters used in the tables: *C* stands for convergence, *D* stands for divergence and *W* is weak convergence. In the brackets, next to the letters is the amount of tests, in which the Mode converged to the correct category.

It is possible to conclude that the Mode 2 is currently the best option for optimization. It seems, that the strictly discrete Mode 1 has problems when the measurement is not proper. This mirrors into the Hybrid sum (Mode 3) and drags its performance down. Although there was divergence in some

	Correct Prior	Incorrect Prior	Uniform Prior
Proper Meas.	4/0/1	2/0/3	0/0/5
Noisy Meas.	0/0/5	0/0/5	0/0/5

Table 4.1: Mode 1 – Virtual simulation, tests are noted in the order: C/W/D.

	Correct Prior	Incorrect Prior	Uniform Prior
Proper Meas.	5/0/0	5/0/0	5/0/0
Noisy Meas.	5/0/0	4/0/1	5/0/0

Table 4.2: Mode 2 – Virtual simulation, tests are noted in the order: C/W/D.

	Correct Prior	Incorrect Prior	Uniform Prior
Proper Meas.	5/0/0	5/0/0	1/0/4
Noisy Meas.	5/0/0	0/0/5	0/0/5

Table 4.3: Mode 3 – Virtual simulation, tests are noted in the order: C/W/D.

cases, it is still possible to extract the `argmax` of the properties and use these against a reference.

4.2 Simulation Experiments

Similarly as in the previous section, the terminology used is the same, but the actual measurement simulation is read from the physical interaction with the object, as opposed to manually creating the measurement based on the reference and perturbing with the random uniform noise. The way MuJoCo simulation creates all of the physically simulated objects can be seen in the Fig. 4.1.

In this section only two materials were used for clear and easy showcase of the action selection. Creating more original materials with clear differences in properties is of great difficulty. The reasons were in detail explained in the Section 3.1.3.

From the previous section, it is apparent that the *Mode 2* is the most useful. That is because of the fact that the *Mode 2* is less likely to be set off course by the measurements with mean values non-corresponding to the ones in the reference bank, in comparison to Mode 1. It is intuitively understandable from the Fig. 4.2, that the two materials are ambiguous in the elasticity and distinguishable in the density. This should intuitively result into the action selection algorithm choosing always the more distinguishable action first. And that is actually the case. This shows the validity of the entropy-based action selection. This observation was made with *Mode 2* in all contexts containing *Good Prior* or *Bad Prior*. However, it needs to be pointed out, that the differential entropy of the Gaussian Mixture changes, based on the interplay between the Material probability distribution (the PMF) and the reference

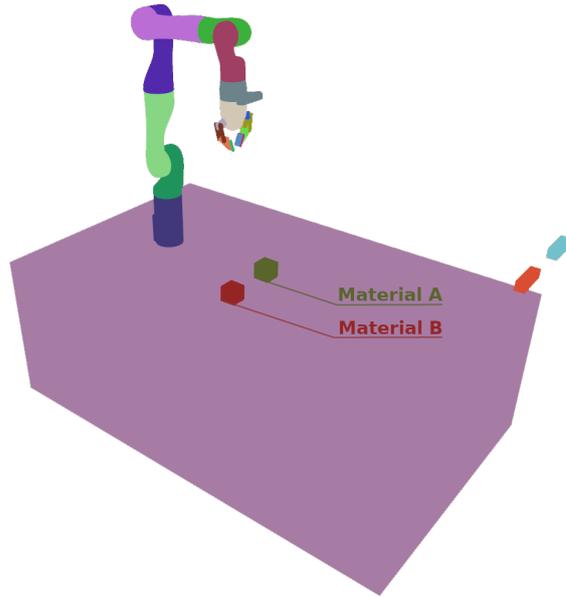


Figure 4.1: A visualization showing all physically-simulated objects as perceived by the MuJoCo engine. Green and Red are materials A and B. Two floating blocks on the right represent the cameras on the real setup.

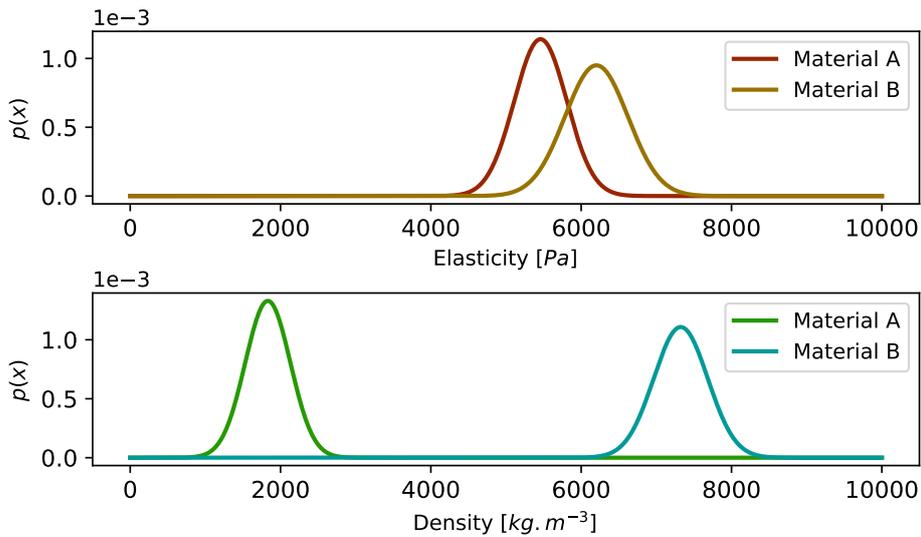


Figure 4.2: Graphical representation of the reference materials A and B for Simulation Experiments.

database (from which, as described in the Section 3.2.5, the Gaussian Mixture Model is constructed). This may, in special cases, alter the planned action. The *Uniform Prior* places the same expectation on both Gaussians in the Elasticity, which renders the Mixture more narrow than in the case of density. In the case of entropy, it is possible to say that the narrower the overall distribution, the smaller the expected entropy. And the smaller the expected entropy, the larger the information gain. The Gaussian Mixtures of both

properties are illustrated in the Fig. 4.3.

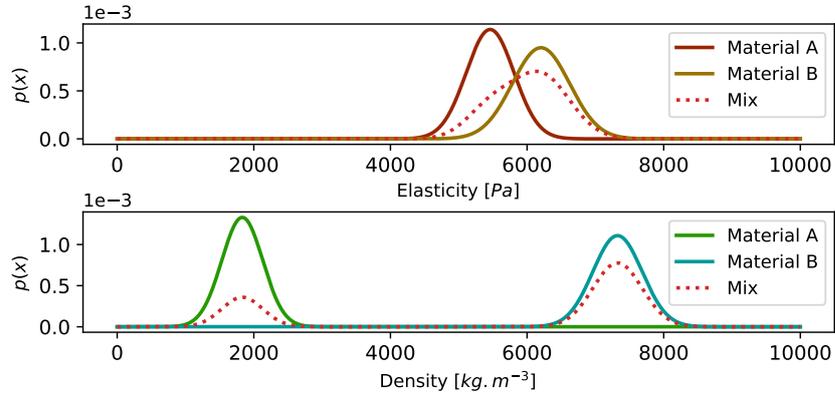


Figure 4.3: The reference materials A and B with the Gaussian Mixture Model shown as *Mix*.

The example mixture from Fig 4.3 corresponds to 0.2763 differential information gain for *Elasticity* and 0.4236 differential information gain for *Density*. The aforementioned prior and measurement context corresponds to the first row and the third column of Tab. 4.4. The algorithm chooses its first action as expected in almost every case, except for the rows 3 and 6, where the uniform prior is used. That is because of the fact, that the context with uniform prior generates the Gaussians in the elasticity property with a relatively same height which is not the case with a non uniform prior. With a non uniform prior the distribution with lower probability *drags down* the other one. On the special occasion of uniform prior and proper measurement, the elasticity Gaussian Mixture Model is narrower (provides more information gain) than the density. Here is the table with the numbering convention of the context again, for easier readability and reference:

1. Correct prior, proper measurement
2. Incorrect prior, proper measurement
3. Uniform prior, proper measurement
4. Correct prior, noisy measurement
5. Incorrect prior, noisy measurement
6. Uniform prior, noisy measurement

The following are the results for the action selection in the simulation. The verdict overall converged to the correct prior everytime. Those measurements are not perturbed with random noise as was the case in the previous section, an error is naturally introduced with the imperfect squeezing in the simulation. That is because of the way the Soft-Bodies are constructed in the MuJoCo simulation. Changing the mass of each capsule in the Soft-Body changes not only the overall weight, but also the net inertia and therefore the resistance against motion. Relating the elasticity, the reference samples were taken

Context	Actions in		
	Mode 1	Mode 2	Mode 3
1	W	W	W
2	W	W	W
3	W	S	W
4	W	W	W
5	W	W	W
6	W	S	S

Table 4.4: Overview of first planned actions by the action selection algorithm in given prior and measurement contexts. The letter *S* stands for *Squeezing* and *W* stands for *Weighing*.

for objects of the same weight, only the **Stiffness** parameter was varied. Therefore having an object of other weight with the stiffness same as in the reference results in a slightly different measurement of Young’s modulus. Such difference does not vary between independent measurement, those are very consistent. The measurements provided are not exhaustive and serve only as a proof of concept. More thorough testing with various squeezing angles and cube sizes and more iterations, is left for further work. The results are shown in the Tables 4.5 to 4.7. The notation is the same as in the previous section.

	Correct Prior	Incorrect Prior	Uniform Prior
Proper Meas.	3/0/0	3/0/0	3/0/0
Noisy Meas.	3/0/0	3/0/0	3/0/0

Table 4.5: Mode 1 – Physical Simulation, tests are noted in the order: C/W/D.

	Correct Prior	Incorrect Prior	Uniform Prior
Proper Meas.	3/0/0	3/0/0	2/0/1
Noisy Meas.	3/0/0	3/0/0	3/0/0

Table 4.6: Mode 2 – Physical Simulation, tests are noted in the order: C/W/D.

	Correct Prior	Incorrect Prior	Uniform Prior
Proper Meas.	3/0/0	3/0/0	2/1/0
Noisy Meas.	3/0/0	3/0/0	3/0/0

Table 4.7: Mode 3 – Physical Simulation, tests are noted in the order: C/W/D.

The results seem promising. Upon closer examination, the good results may be caused by the small variable space (i.e. *Material A*, *Material B* and the *Joker*), and the relatively large σ_{ref} of those materials. In the section before, the mean of the measurement was randomly perturbed so the Mode 1 usually diverged which is not the case here. Although in this simulation, regarding the Modes 2 and 3 in the cases with Uniform prior, the action

selection is occasionally prone to error. Although, the statistical sample of the test bank is not large enough to rule out potential statistical flukes. The automation of the testing process was inhibited due to seemingly random ambiguous trajectories of the robotic manipulator that sometimes throw the object away due to centrifugal force while rotating. Such measurements had to be manually scrapped, because the results of such test automatically diverged and were not distinguishable from the real divergence caused by an actual measurements.

4.3 Real-World Experiments

To conclude the series of action selection tests, real-world experiments are conducted with the Kinova mainpulator. For detailed setup description see Section 3.1.4. A series of experiments similar to those in the pveious subsections is done. However, tests in this section lack the division into contexts with a *Proper* or *Noisy* measurement, because the actual measurement from the manipulator is used as is. That means that only three contexts are now used, which are:

1. Correct prior
2. Incorrect prior
3. Uniform prior

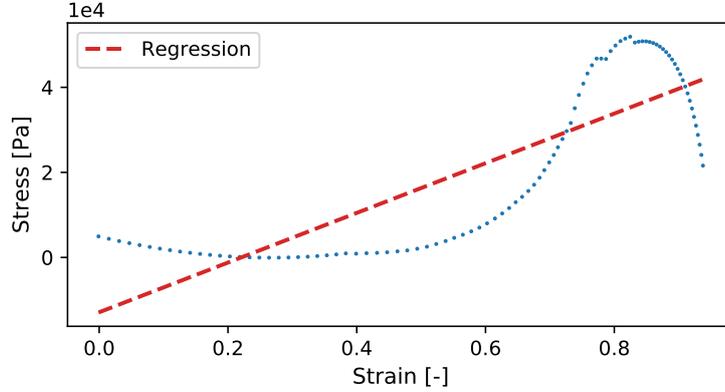
The definitions of the prior correctness is in accordance with previous sections. The sigma of density measurement was estimated as an average of material standard sample deviations from the Table 3.2. The sigma for elasticity measurement with the gripper Robotiq 2F-85 was obtained by evaluating a set of measurements with deformable objects (provided by Shubhan Patni and Matěj Hoffmann). The sigma values are shown in the Table 4.8. For

Property	Density [kg m^{-3}]	Elasticity [kPa]
Meas. Error	$\sigma_D = 29.64$	$\sigma_E = 4.091$

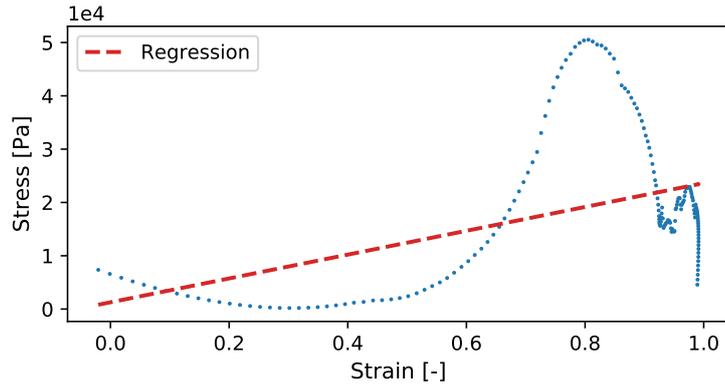
Table 4.8: Real-world measurement errors for properties.

each Mode and each Prior type (i.e. *Correct*, *Incorrect* or *Uniform*), 2 tests consisting of 3 algorithm steps were done, where each measurement was taken as an average of 3 measuring samples of the object. Weighing action is remarkably precise in comparison with the squeezing. The Young’s Modulus is estimated as the slope of a line, which is linearly regressed on the measured data. This approximately serves as a Young’s modulus estimation. The data processing part was provided by another IPALM member, *Shubhan Patni*. There are a few different ways to extract the Young’s modulus from the time series data, such as tangent modulus, maximum slope method or chordal modulus [32, p. 17]. Due to the technical restrictions of the gripper, linear regression is deemed as the most useful. Data is first filtered to smooth out the motor current spikes from the inner controller of the gripper. However,

not all datapoints can be filtered efficiently without destroying too much information from the times series data—see Fig. 4.4(a) for correct and (b) for incorrect estimation. Because of such limitations, more than one elasticity



(a) : Accurate measurement without excess effort points caused by the gripper’s controller. The linear regression of Young’s modulus here is 17 kPa.



(b) : Noisy measurement with excess effort points caused by the gripper’s controller. The regression of Young’s modulus here is 0.49 kPa.

Figure 4.4: Visualization of Young’s modulus estimation.

measurement was needed to decrease the impact of such errors. The Robotiq 2F-85 gripper is designed for stable grasp, not for incremental measurement. In contrast with the Kinova joints, the gripper lacks torque sensors. The results of conducted tests can be seen in the Tab. 4.9. The imbalanced

	Correct Prior	Incorrect Prior	Uniform Prior
Mode 1	2/0/0	2/0/0	2/0/0
Mode 2	2/0/0	0/0/2	0/1/1
Mode 3	2/0/0	0/0/2	0/0/2

Table 4.9: Results for all Modes, tests are noted in the order: C/W/D.

precision of weighing and squeezing reflects heavily to the testing process.

The Mode 1 successfully converged to the correct category in 6 out of 6 tests. That is mostly because of the action selection algorithm choosing the weighing action every time. That is intuitively correct, given that the objects are very similar in the elasticity and distinguishable in the density, see Fig. 4.5. The Mode 2 failed in the settings where the prior is different from the measurement. After the first measurement update (which is almost every time the density—see Tab. 4.10) squeezing action is chosen, which produces imprecise measurements and causes the classification to fail. With only two actions available, this means that the action selection provides proper classification only with weighing measurements in the scope of three steps of the algorithm.

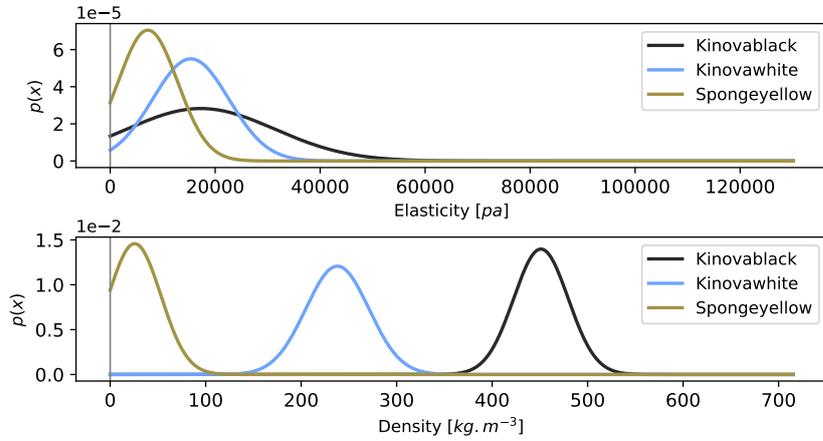


Figure 4.5: Reference of the real-world objects. *Note:* this figure is also displayed in the Section 3.1.4. It is also placed here for the reader’s convenience.

Mode	Prior		
	Correct	Incorrect	Uniform
Mode 1	W	W	W
Mode 2	W	W	S
Mode 3	W	W	W

Table 4.10: Overview of first planned actions by the action selection algorithm in given prior contexts. The letter *S* stands for *Squeezing* and *W* stands for *Weighing*.

The squeezing is chosen only in the Mode 2 (only differential information gain) with Uniform prior. This is due to the same reasons as explained in the previous Section 4.2. It is only a matter of how far are the mean values in the property from each other, how great are their deviations and also how much belief is put into the categories in question. The real-world tests serve mostly as a proof of concept. To rule out potential statistical flukes, a more thorough testing with more varied Action and Variable space (e.g. friction estimation developed by other IPALM members [9]) will be necessary.

4.3.1 Results

From the metric used, the Mode which brings the best classification results is the Mode 2 with 88.89% accuracy. Mode 3 and Mode 1 follow with 64.81% and 55.56% accuracy respectively. The accuracy is computed as the number of successful classifications over the overall number of classifications throughout all environments (i.e. over *Virtual*, *Simulation* and *Real-World* environments, to arrive at a most general classification metric possible). The metric is summarized in Tab. 4.11. The importance of each testing environment is

	Accuracy
Mode 2	88.89 %
Mode 3	64.81 %
Mode 1	55.56 %

Table 4.11: Overview of the different Mode accuracy results.

naturally weighted by the amount of tests done in such environment. For clarity, the Virtual setup is assumed to give greater insight into the validity of action selection steps, therefore a greater number of tests are accounted for which naturally biases the accuracy calculation in favor of the Virtual environment testing, and therefore in favor of the optimization Mode 2. It is important to repeatedly note, that the second Mode chooses actions, which generate measurements that update the prior belief. The Modes simply differ in the way of computing entropy and consequently information gain. The reason that the Mode 2 has the best results probably stems from the fact that it takes into account the object in greater complexity—evaluates the entropy of the properties. The Mode 1 does not provide such possibility and often chooses the wrong action only optimizing for the larger information gain in the categories. The Mode 1 may then intentionally choose the action that would, for example, choose the more noisy action, because it promises to bring greatest information gain, although this often means choosing the unknown category—*joker*.



Chapter 5

Conclusion

In this thesis, an algorithmic approach based on information gain from discrete and continuous information entropy is proposed, implemented and tested. Such algorithm may find use in interactive settings, where the environment is changing constantly and the agent needs to gather and update information all the time. Choosing the action with the most learning potential accelerates information gathering and provides more information overall. In this thesis a situation simulating the real world is created, where the goal for the manipulator is to gather information about the object properties and/or category. The algorithm is comprised of the prior evaluation via entropy, measurement emulation based on expected measurement error, expected information gain calculation from information entropy estimation and finally Bayesian update after conducting the actual measurement. In this thesis the whole algorithm is explained in theoretical detail and implementation is available at [10]. Three optimization Modes are proposed. Each of those Modes is tested in Virtual, Simulated and Real-World environment. Each testing environment brings its own advantages and disadvantages. The Virtual environment allows for quick evaluation, but lacks the specific problems that arise in the actual Real-World testing. On the other hand, it can be argued that the absence of the technical difficulties allows for a clearer, less biased, evaluation of the action selection process.

The Simulation environment, as opposed to the Virtual environment, is better in revealing implementation problems, such as problems with the actual grasping, kinematics and other object-gripper interactions that are not accounted for in the latter. Simulating physical contacts comes with its own technical setbacks, such as numerical instabilities, unwanted Soft-Body object penetration, and other simulational inconsistencies.

The Real-World is the desired implementation environment. Such environment brings measurement imperfections. It is hard to conclude how much do the measurement errors cloud the judgment of the action selection algorithm, because the classification metric used is only a proxy. The resulting classification is inherently dependent on the measurement precision, which may in conclusion return wrong classification even though the right action is chosen. The simulational difficulties, real-world problems with measurements and other related topics are discussed in the next section.

Chapter 6

Discussion and Future Work

Throughout the work on this thesis, plenty of complications and technical difficulties occurred, mainly concerning the simulation environment in MuJoCo. The MuJoCo physics engine is very complex and allows to apply all sorts of conditions and restrictions on the behavior of the Soft-Body objects. Majority of such work was out of the scope of this thesis, although it would probably prove very useful. The main problem was the stability of squeezing simulation, which depended on the mass of the individual Soft-Body capsules, the stiffness of the tendons and joints, the spacing between the capsules and conditional restrictions of the free dimensions. All of this interacts with the simulation of friction. Aforementioned points complicate the assembly of a larger database which could be then used for reliable simulating of robotic interaction with the Soft-Body objects. The database used in the Section 4.2 is too small to account for more complex prior-measurement contexts, which would test the performance of the action selection algorithm into greater depth. With the current database, only a distinction between the two objects, which were ambiguous in one property and distinguishable in the other, could be made. On the other hand, the action selection was able to reliably choose the intuitively optimal action, as it chose the action which would discriminate the objects to a greater extent.

The real-world measuring apparatus in the form of the Kinova manipulator and 2F-85 gripper are very user-friendly for operation. Although the feedback from the gripper is not designed for such stiffness estimations, nevertheless it is used. In everyday robotics, it is not reasonable to expect having professional tools available at all times for the manipulator. In this thesis, it is shown that even a general-use manipulator can be useful for object exploration and possibly classification. This paves the way to more independent robotic manipulation and exploration.

The action selection algorithm provides the desired metric for choosing the optimal action. When the classification *diverges* as explained previously, it is actually a convergence to an unknown category, called *joker*. This also provides valuable information that the measurements are not on par with the database. That effectively lowers the overall uncertainty in the category. If such case occurs, argmax -information on the maximum argument of the Gaussian Mixture Model—is provided by the action selection pipeline in the

measurement logs for both properties and could be further utilized [10]. The size of the Bayesian network (see Fig. 3.15(b)) does not yet call for larger inference models. Using large inference models utilizing frameworks such as STAN or PyMC proved to be quite difficult and was not yet needed for the scale of the current Bayesian network. Although with increasing size of the Action and Property repertoire, a larger inference model would be beneficial as it could also provide message passing and parameter regression of the multinomial distributions automatically via MCMC algorithms and so on.

6.1 Future Work

Complex grasping, shape and friction exploration

In this thesis, all objects grasped and squeezed were homogeneous cubes. Gripping of such objects is straightforward. If grasp is successful, measurements such as squeezing or weighing can take place. However if the object is of a more complex shape (e.g. mug, bowl) the grasp can prove to be difficult. For such cases an exploratory grasping sequence may be implemented, which would combine visual information from the robot's on-body camera and the tactile information from the gripper as suggested by Nikandrova et al. [6]. In this setting, more complex grasps may be executed. Additional information for a better grasp may come from a friction map estimation and grasping in the areas of high friction density, as proposed by Nguyen Le et al. [9] Such grasping information pipeline would be beneficial for the action selection algorithm described in this thesis, as the usefulness of the action repertoire would rise.

Message Passing and Statistical Modeling

Regarding the Bayesian network in this thesis, the current belief is propagated only in the parent-child pair in each measurement episode. There is currently no message-passing in between the child nodes. A message-passing would help with providing information from one property among others, resulting in a more whole belief about the explored object. For such message passing a statistical modeling framework would be needed to handle all of the informational interaction.

Reliable MuJoCo Soft-Body Simulation Dataset

MuJoCo provides a great amount of settings for simulating the physics of Soft-Body objects. The potential of the physics engine was not completely fulfilled in this thesis. Simulation of composite Soft-Body objects, which are not only box or ellipse shaped should be possible. This could lead to simulation of the manipulation and stiffness of more complex bodies of non-trivial shape. Such simulation would provide important insight into testing of different grasps and overall manipulations. It would also provide a larger database of

Soft-Body objects that would allow for the action selection algorithm to be more thoroughly tested.

■ Utilizing directly Information from a Visual prior

In this thesis, prior probabilities were chosen *ad hoc*, depending on the *prior-measurement* context needed for action selection testing. The original plan was to use prior probabilities from the Object Category and Material inference [30]. However, the image inference network is not completely deployment-ready and provides only a shape-material dependent information. The neural network implemented in the mentioned project is called *detectron2* and provides only limited prior information. The information on material is dependent on category and vice versa. The subsequent work could provide independent category and material prior, because the current inference is biased. If some cups used for training are usually made of ceramic and are white, the output will be `cup:ceramic`. However, this does not infer the material directly but from the information that cups are usually ceramic. Nevertheless, setting the initial prior is not essential for the action selection algorithm, which is why it is left for further work.

■ Alternative performance measures for the action selection algorithm

Another way to measure the performance of the action selection algorithm would be to compare the information gain with random action selection as a baseline. With the current setup with only two actions, it requires vast amounts of measurements to extract any useful information about the comparison with random agent. Against random agents, the odds are 50:50 in this case. Such comparison would make sense provided that the manipulator action repertoire comprises of at least 3 or more actions. This alternative performance measure is left for further work, until there are more actions available.



Bibliography

- [1] Thomas Duemmler, Petra Schoeberl, and Gudrun Schwarzer. Development of visual center of mass localization for grasp point selection. *Cognitive Development*, 23(3):370–384, 2008.
- [2] Susan Lederman and Roberta Klatzky. Extracting objects’ properties by haptic exploration. *Acta psychologica*, 84:29–40, 11 1993.
- [3] Interactive perception-action-learning for modelling objects (IPALM). <https://sites.google.com/view/ipalm>, 2021. [Online; accessed 12-April-2021].
- [4] J. Felip, J. Laaksonen, A. Morales, and V. Kyrki. Manipulation primitives: A paradigm for abstraction and execution of grasping and manipulation tasks. *Robotics and Autonomous Systems*, 61(3):283–296, 2013.
- [5] Adam Spiers, Minas Liarokapis, Berk Calli, and Aaron Dollar. Single-grasp object classification and feature extraction with simple robot hands and tactile sensors. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5073–5080, 10 2015.
- [6] Ekaterina Nikandrova, Jonna Laaksonen, and Ville Kyrki. Towards informative sensor-based grasp planning. *Robotics and Autonomous Systems*, 62(3):340–354, March 2014.
- [7] Jack Collins, Shelvin Chand, Anthony Vanderkop, and Gerard Howard. A review of physics simulators for robotic applications. *IEEE Access*, PP:1–1, 03 2021.
- [8] Sean McGovern, Huitan Mao, and J. Xiao. Learning to estimate centers of mass of arbitrary objects. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1848–1853, 2019.
- [9] Tran Nguyen Le, Francesco Verdoja, Fares J. Abu-Dakka, and Ville Kyrki. Probabilistic surface friction estimation based on visual and haptic measurements. *IEEE Robotics and Automation Letters*, 6(2):2838–2845, 2021.

- [10] A. Kruzliak. Action-selection algorithm, gitlab root folder. <https://gitlab.fel.cvut.cz/body-schema/ipalm/ipalm-action-selection>, 2021.
- [11] A. Kruzliak. Action-selection algorithm, complementary content. <https://drive.google.com/drive/folders/1vQ8vQrWrMCKUBtV1YpBp2vnnc8HAgTI6?usp=sharing>, 2021.
- [12] Mahen Mahendran. The modulus of elasticity of steel - is it 200 gpa? *International Specialty Conference on Cold-Formed Steel Structures*, 5, Oct 1996.
- [13] Michele Dondi, Ercolani G., Marsigli M., Melandri C., and Claudio Mingazzini. The chemical composition of porcelain stoneware tiles and its influence on microstructure and mechanical properties. *InterCeram: International Ceramic Review*, 48:75–83, 01 1999.
- [14] Tomáš Húlan and Igor Štubňa. Young’s modulus of kaolinite-illite mixtures during firing. *Applied Clay Science*, 190:105584, 2020.
- [15] Douglas C. Giancoli. *Physics: principles with applications, 7/E*. Pearson, 2014.
- [16] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.
- [17] M. Pliska. Squeezing of deformable objects in MuJoCo simulation. <https://drive.google.com/file/d/1cciFhDajglJrfr9wDQ8IdhIjIajQyFED/view?usp=sharing>, 2021.
- [18] Mujoco – xml model reference. <http://www.mujoco.org/book/XMLreference.html#composite>.
- [19] A. Koubâa. Robot operating system (ROS): The complete reference (volume 1). 2016.
- [20] J. Behrens. Kinova MuJoCo simulation. https://github.com/JKBehrens/kinova_mujoco, 2021.
- [21] Kinova gen3, technical resources. <https://www.kinovarobotics.com/en/resources/gen3-technical-resources>.
- [22] Robotiq 2f-85 gripper, technical resources. https://assets.robotiq.com/website-assets/support_documents/document/online/2F-85_2F-140_TM_InstructionManual_HTML5_20190503.zip/2F-85_2F-140_TM_InstructionManual_HTML5/Content/6.%20Specifications.htm.
- [23] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.

- [24] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.
- [25] Kevin P. Murphy. Conjugate bayesian analysis of the gaussian distribution. Technical report, 2007.
- [26] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.
- [27] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.
- [28] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [29] The ycb object dataset. <https://www.ycbbenchmarks.com/object-set/>.
- [30] J. Hartvich, A. Kružliak, and M. Pliska. Object category and material inference with detectron2. <https://prezi.com/i/rrf1sj4fgrxr>, 2020.
- [31] J. Hartvich, A. Kruzliak, and M. Pliska. Object category and material inference, code library. <https://gitlab.fel.cvut.cz/body-schema/ipalm/ipalm-vir2020-object-category-from-image>, 2021.
- [32] J D Lord and R M Morrell. Elastic modulus measurement—obtaining reliable data from the tensile test. *Metrologia*, 47(2):S41–S49, mar 2010.

I. Personal and study details

Student's name: **Kružliak Andrej**

Personal ID number: **483496**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Exploratory Action Selection to Learn Object Properties Through Robot Manipulation

Bachelor's thesis title in Czech:

Výběr průzkumných akcí za účelem zjištění vlastností předmětů robotickou manipulací

Guidelines:

1. Develop a probabilistic representation of object attributes such as category, material, stiffness/elasticity, mass, etc., using a combination of discrete and continuous probability distributions. The focus is on physical object properties that are difficult or impossible to extract from visual information only.
2. Set up dependencies between the variables using a Bayes net.
3. Develop a framework for inference in the network when a measurement of an object attribute arrives, possibly using frameworks such as STAN (<https://mc-stan.org/>) or <https://docs.pymc.io/>. Measurements will be manipulation actions using a robot hand such as lifting or squeezing the object.
4. Calculate the uncertainty over all object attributes (e.g., using entropy) and prepare an action selection algorithm - choose the action that is likely to reduce the uncertainty the most.
5. Connect the framework to a database of objects and their attributes, in collaboration with partners from project IPALM (<https://sites.google.com/view/ipalm>).
6. Test the framework in a simulation environment (Kinova Gen3 manipulator with Robotiq 2F-85 gripper in Mujoco, ROS & MoveIt!). A basis of this environment is already developed.
7. Conduct and evaluate experiments on a real robot (Kinova Gen3 + Robotiq 2F-85). Priors about object category and material from image will be acquired using <https://gitlab.fel.cvut.cz/body-schema/ipalm/ipalm-vir2020-object-category-from-image>.

Bibliography / sources:

- [1] Felip, J., Laaksonen, J., Morales, A., & Kyrki, V. (2013). Manipulation primitives: A paradigm for abstraction and execution of grasping and manipulation tasks. *Robotics and Autonomous Systems*, 61(3), 283-296.
- [2] Liarokapis, M. V., Calli, B., Spiers, A. J., & Dollar, A. M. (2015, September). Unplanned, model-free, single grasp object classification with underactuated hands and force sensors. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5073-5080). IEEE.
- [3] Nikandrova, E., Laaksonen, J., & Kyrki, V. (2014). Towards informative sensor-based grasp planning. *Robotics and Autonomous Systems*, 62(3), 340-354.
- [4] McGovern, S.; Mao, H. & Xiao, J. (2019), Learning to estimate centers of mass of arbitrary objects, in 'Intelligent Robots and Systems (IROS), 2019 IEEE/RSJ International Conference on', pp. 1848-1853.
- [5] Ottenhaus, S., Kaul, L., Vahrenkamp, N., & Asfour, T. (2018). Active tactile exploration based on cost-aware information gain maximization. *International Journal of Humanoid Robotics*, 15(01), 1850015.
- [6] Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence-A Modern Approach*, Third International Edition. Chapter 14: Probabilistic Reasoning.

Name and workplace of bachelor's thesis supervisor:

Mgr. Matěj Hoffmann, Ph.D., Vision for Robotics and Autonomous Systems, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Jan Kristof Behrens, MSc., Robotic Perception, CIIRC

Date of bachelor's thesis assignment: **12.01.2021** Deadline for bachelor thesis submission: **21.05.2021**

Assignment valid until: **30.09.2022**

Mgr. Matěj Hoffmann, Ph.D.
Supervisor's signature

prof. Ing. Tomáš Svoboda, Ph.D.
Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature



Appendix A

Action selection – Virtual

The attached file `kruzland-action-sel-virtual.zip` contains the basic software for the virtual setting for the Action selection algorithm. For more information refer to the `README.md` file. The other contexts of actions are above the attachment size limit and are located at the previously provided GitLab repository [10].