Human-Assisted Model Identification of Articulated Objects

Scientific work to obtain the degree
Bachelor of Science (B.Sc.)

at the Human-centered Assistive Robotics
Technical University of Munich

Submitted by Paul Delseith
on 07. 09. 2021

First supervisor: Univ.-Prof. Dr.-Ing. Dongheui Lee

Second supervisor: Christoph Willibald

TECHNISCHE UNIVERSITÄT MÜNCHEN

Human-centered Assistive Robotics

UNIV.-PROF. DR.-ING. DONGHEUI LEE

August 30, 2021

BACHELOR THESIS
for
Paul Delseith
Student ID 03710332, Degree EI

Human-Assisted Model Identification of Articulated Objects

Problem description:

A kinematic model of articulated objects in the robot's surrounding can improve the robot's confidence when interacting with the environment during complex manipulation tasks. Many works have been conducted that enable autonomous execution of predefined tasks using models of articulated objects [1]. Identifying the kinematic structure of the robot's environment typically requires expert knowledge and consumes a lot of time. Except for a few approaches [2], the problem of automatically identifying kinematic models of articulated objects has not been investigated in depth in current literature.The number of joint parameter combinations grows exponentially with the length of the kinematic chain, which is a major difficulty for efficiently and reliably identifying a kinematic model. This work therefore focuses on employing non-expert human operators to reduce the search space for semi-automatic model identification for articulated objects by manual exploration of the environment. The human should be able to intuitively collect data and to share his intuition about the manipulated objects. Therefore the following tasks have to be conducted:

Tasks:

- Literature research on model identification of articulated objects
- Development and implementation of a prototypical handheld device that allows an operator to record data while interacting with the objects in the environment
- Development of a GUI that allows an operator to share additional insights about the objects during data acquisition
- Implementation of a signal processing strategy for noise reduction and sensor fusion
- Implementation of a model fitting algorithm based on collected sensor data from simulation
- Evaluation of the semi-automatic model identification on real hardware

Bibliography:

[1] Adrian Röfer, Georg Bartels, and Michael Beetz. Kineverse: A symbolic articulation model framework for model-generic software for mobile manipulation. *arXiv preprint arXiv:2012.05362*, 2020.

[2] Jürgen Sturm. *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots*. PhD thesis, PhD thesis, University of Freiburg, Germany, 2011.

Supervisor:	M. Sc. Christoph Willibald
Start:	25.05.2021
Intermediate Report:	29.06.2021
Delivery:	12.10.2021

(D. Lee)
Univ.-Professor

Abstract

In this work, I propose a framework for human-assisted identification of articulated objects. An articulated object could be any mechanism in the environment we humans interact with to change its state, like a door or drawer. The system developed in this thesis allows a human operator to identify the kinematic structure of an articulated object through a guided measurement procedure. Informed by examples in recent literature, it is assumed that by modeling articulated objects as a series of interconnected joints, planning for sophisticated autonomous robot interaction with these objects is already possible. Therefore, the identified models only include information on the associated joints, their pose in 3D space, their parameterizations, and their overall connectivity and do not model the full 3D shape of an object. A measurement application is developed that instructs an operator to interact with the different joints of an articulated object one by one, such that joint models for each can be identified from the movement data. Movement data is collected through marker-based motion capture that tracks a self-built gripper which the operator uses to articulate the mechanism. Tests are conducted in real-world experiments to evaluate whether the devised procedure allows accurately identifying single joint models. These tests show that the system implemented in this thesis enables the accurate identification of single joint models and complete kinematic models of articulated objects. However, there are still problems both with the mechanics of the data collection and the underlying procedure used to fit joint models that negatively influence accuracy and ease of use. These problems mainly occur when interacting with small, delicate mechanisms and handles. Identified models can be queried for joint and handle poses given a joint configuration vector. This way, they could be used in planning tasks.

Contents

Chapter 1

Introduction

Most interesting and relevant robotics tasks involve robots interacting with mechanisms in their environment, opening doors, drawers, picking and placing objects. Humans interact with articulated objects all the time. Understanding these mechanisms and developing efficient control schemes for manipulation is crucial for building autonomous robots that confidently navigate human-dominated domains. Although there are control-based approaches that enable robots to interact with these objects without a specific mechanism model [JK09], these are limited to simple interactions that only involve a single movement. Furthermore, there is still a need for human supervision to tell the robot where an object can be grasped for manipulation, and how to start the interaction. By presenting robots with knowledge of the underlying models, advanced planning procedures, considering the constraints imposed by the models, can be employed [BHB13, CCL10, JN19, RBB20, RSP$^+$12]. Even when the underlying models are given, it is still a long way to achieving human performance and versatility in these articulation tasks. However, we see it as the most promising approach to solving this problem. In recent years, research has addressed the topic of (semi-)automatically identifying models for articulated objects on multiple occasions [Stu11, MMB17, JN19]. However, these approaches are still relatively rigid in what attributes are modeled and how these are learned. The most important information about an articulated object is certainly its kinematic structure. However, there are many more interesting attributes associated with articulated objects. For some mechanisms, these could be crucial for successful interaction. Furthermore, they could enable sophisticated reasoning about an articulated object's state.

This thesis aims to design and build a proof of concept system for identifying models of articulated objects through user demonstration. Unlike previous approaches, where the human operator serves as a quick, convenient way to gather articulation data, the proposed approach should consider the human operator a central component in the model identification procedure. Therefore, an interactive measurement application should be implemented that guides the user through the identification process, computes model fits in real time, visualizes them, and allows the user to enter insights about the mechanism and intervene when there are obvious problems. This work focuses on implementing a basic approach for identifying the kinematic structure of articulated objects. It aims to verify whether there is merit to this approach. This work should then serve as a base for a modular system for detailed, human-guided modeling of articulated objects.

1.1 Contributions

The main contribution of this work is a proof of concept system for human-assisted identification of kinematic models for articulated objects. We propose a guided interactive measurement procedure. All current approaches either make strong assumptions about the objects they try to identify (like that they only have one joint) or need simultaneous information about the movement of all parts of an articulated object to find a model. In contrast, the method devised in this work should only use end effector pose information to find models of a broad range of articulated objects. The missing sensory information or model assumptions should be replaced by insights of the human operator about the structure and functionality of the object. Using human insights directly in the model fitting approach rather than running a big optimization makes the model fitting more transparent. Therefore structural errors should be easier to identify. Furthermore, directly considering the human operator in the model identification makes it possible to add new model parameters quickly. The operator has immediate intuitive insights about any mechanism and can quickly learn new measurement procedures.

Chapter 2

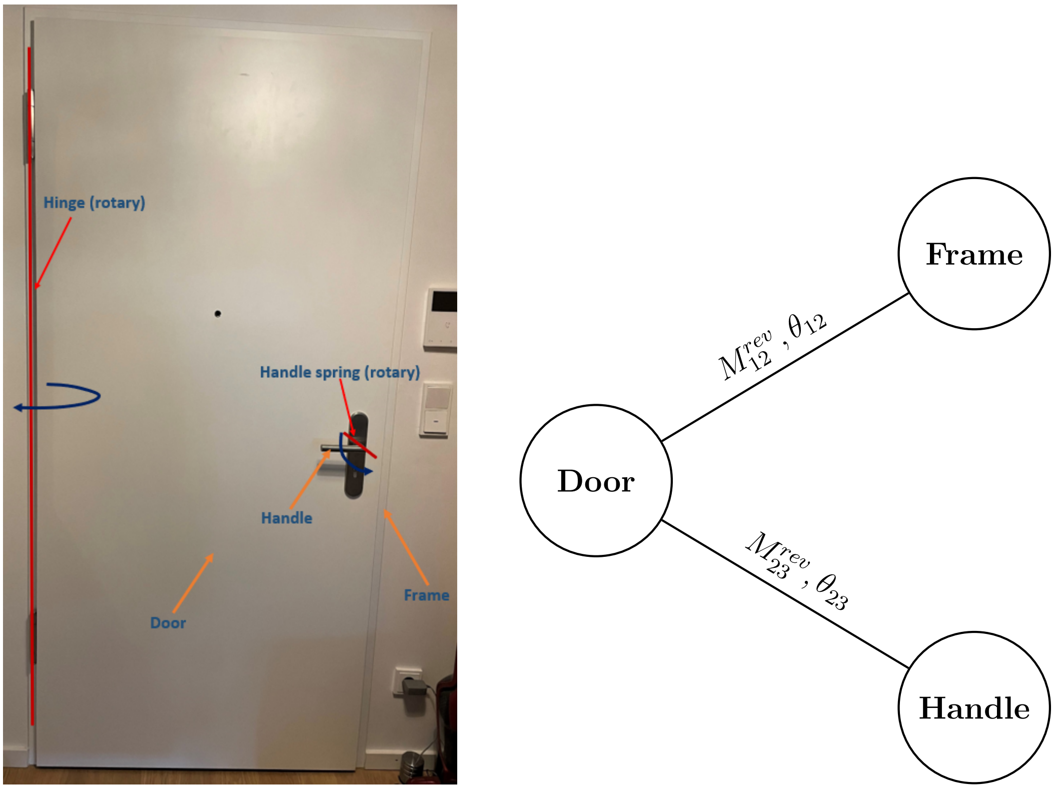
Related Work

In this section, I would like to give a broad overview over the field of model identification of articulated objects. I split the problem up into two domains, identification of kinematic and identification of dynamic attributes, and will subsequently show the progress made in both areas. Furthermore, to prove that these models can actually be used in robotic manipulation, I also present some approaches that employ learned models for autonomous robotic interaction with articulated objects.

2.1 Identification of Kinematic Models

The problem of identifying kinematic models for simple everyday articulated objects has been approached in the literature on multiple occasions. A central approach often referenced in consecutive works is presented by Sturm et al. [Stu11, SSP$^+$09]. The major contribution of this research is a probabilistic description of the problem that enables accurate and efficient decisions on the kinds of joints and connectivity of a given articulated object. The proposed framework uses 6-DoF motion tracking of every rigid body part of an articulated object. In this context, a rigid body represents a body part of an articulated object that always moves as a whole and is connected to other rigid body parts by a joint.

Articulated objects are described as kinematic graphs $G = (V_G, E_G)$ with nodes $V_G = 1, ..., p$ that describe the rigid bodies of the articulated object and edges $E \subset V_G \times V_G$ that describe the kinematic relationship between two nodes respectively. If two nodes are connected by a joint, one of four considered joints can explain their kinematic relationship. The joint models include three standard models, rigid, prismatic, rotary, and a fourth parameter-free model that uses Gaussian process regression [RW05] to learn a joint model that is not restricted in DoF or trajectory shape. This way, the considered models can cover a broad range of mechanisms, as the standard models can explain most regularly encountered joints, and the Gaussian process model can learn most other relationships. An example kinematic graph for a door can be seen in Fig. 2.1, θ_{ij} represents the joint articulation.



(a) Generic door, rigid bodies marked with orange arrows, joints marked with red arrows

(b) Kinematic graph of the door

Figure 2.1: Kinematic structure of a door

Central to the approach is the insight to model the kinematic graph as a Bayesian network where every edge can be described with an observation model. The observation model describes the dependencies between the underlying joint model, model parameters, joint configurations, and observations. An example is shown in Fig. 2.2. During the model fitting, a human articulates the object while a camera system captures the rigid body poses using visual markers. At first, both the joint types and connectivity structure are unknown. The task is split up into two sub-tasks, first, finding the best joint model for every possible combination of two rigid bodies; second, identifying the real connectivity by comparing all possible joints to find the most likely spanning tree. Joint models are fit by optimizing the data likelihood of the observation sequence given the considered model. The data likelihood can be computed based on the observation model seen in Fig. 2.2, where x_i/x_j are the true poses of rigid body i and j, y_i/y_j are the respective observed poses, q_{ij} is the (latent) joint configuration, M_{ij} is the considered joint model with parameter vector θ_{ij}, Δ_{ij} represents the true transformation between the rigid bodies, and z_{ij} represents the observed transformation. We can see that z_{ij} is directly dependent on Δ_{ij}, therefore,

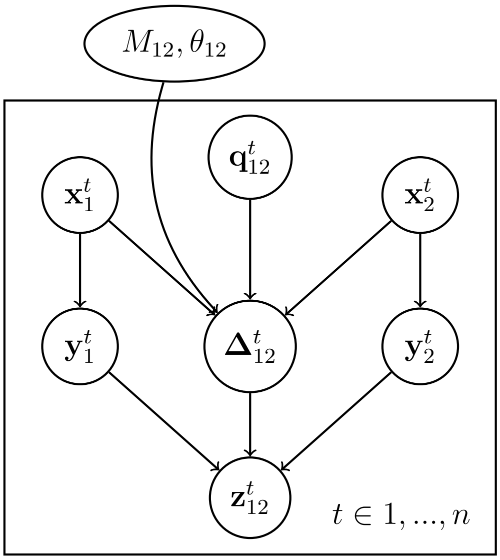


Figure 2.2: Full graphical model of an articulated object consisting of two bodies as used by Sturm in [Stu11]

knowing z_{ij} and assuming a model for the relationship between Δ_{ij} and z_{ij} we can infer the likelihood of z_{ij} given Δ_{ij}. Pose data from a camera-based tracking system can be noise and outlier-prone, so the measurements include noisy inliers and some outliers that the underlying model cannot explain. To capture the underlying prior on the data, the relationship between observed and true transformations is modeled as a mixture of Gaussian distributed inliers and uniformly distributed outliers with a mixing constant γ representing the outlier ratio. For every possible joint, an optimized set of parameters is computed for every possible joint model by maximizing the data likelihood.

Next, the best model for every possible joint has to be selected. This is done using the Bayesian Information Criterion (BIC) [Sch78] (2.1) as a heuristic. Here, $k \in \mathbb{N}$ represents the number of parameters of the current model, and $n \in \mathbb{N}$ represents the number of measurements. The BIC approximates model evidence by weighing data likelihood (first term) and model complexity (second term). The model with the lowest BIC then represents the least complex model that still accurately explains the data.

$$\log p(M|D) \approx \log p(M|D, \hat{\theta}) - \frac{1}{2} k \log n \tag{2.1}$$

In the next step, the found models are used to estimate the overall connectivity of the articulated object. The articulated object is described as an undirected, fully connected graph with the object part poses as nodes and the object links and associated joints as edges. When the considered types of articulated objects are restricted to kinematic trees only, the overall connectivity of a given articulated object will be given by a spanning tree of the graph. As the number of possible spanning trees grows exponentially with the number of nodes, it quickly becomes infeasible to compute and reason about each of these spanning trees. Therefore, a heuristic is employed to find the spanning tree that maximizes the posterior of the kinematic structure for the given data. By only dealing with kinematic trees, independence of the individual links can be assumed, which permits writing the overall posterior as the product of the local posteriors, with $\hat{E}_G$ as the kinematic structure that maximizes the posterior probability, D_z the measurement data, $\hat{M}_{ij}$ the "best" model for the joint connecting nodes i and j, and $\hat{\theta}_{ij}$ the "best" parameter vector we get,

$$\begin{aligned}
\hat{E}_G &= \operatorname*{arg\,max}_{E_G} p(E_G|D_z) && (2.2)\\
&= \operatorname*{arg\,max}_{E_G} p(\{(\hat{M}_{ij},\hat{\theta}_{ij})|(ij) \in E_G\}|D_z) && (2.3)\\
&= \operatorname*{arg\,max}_{E_G} \prod_{(ij)\in E_G} p(\hat{M}_{ij},\hat{\theta}_{ij}|D_z) && (2.4)\\
&= \operatorname*{arg\,max}_{E_G} \sum_{(ij)\in E_G} \log p(\hat{M}_{ij},\hat{\theta}_{ij}|D_z) && (2.5)
\end{aligned}$$

By decomposing the problem into a sum of the local posteriors for each joint (2.5), it can efficiently be solved by finding the minimal spanning tree in an undirected graph with edge costs set as the negative logarithmic posterior (2.6).

$$cots_{ij} = -\log p(M_{ij}, \theta_{ij}|D_{z_{ij}}) \qquad (2.6)$$

This cost is approximated using the Bayesian information criterion (2.1). The approximated kinematic structure can then be identified as the minimal spanning tree. In consecutive work [SJS$^+$10], the framework is expanded to find kinematic models of single-joint articulated objects and use them for real-time manipulation. In this approach the only data available for the joint identification is the robot's end effector pose. The results are fairly accurate and show that this method enables robots with minimal sensory equipment (joint encoders) to almost (starting pose and direction given) autonomously learn models of simple articulated objects. Multiple consecutive works build on the probabilistic framework presented by Sturm et al. adding different ways of acquiring the data needed for identifying the kinematic models [PWT15, HNOS15, RSP$^+$12]. These works mainly focus on computer vision methods, as vision-based marker-less methods allow them to operate in less clean/controlled environments. However, entirely missing from these approaches and the approach as presented by Sturm, is the consideration that the underlying articulation models are often not static, but rather change over time depending on the configuration.

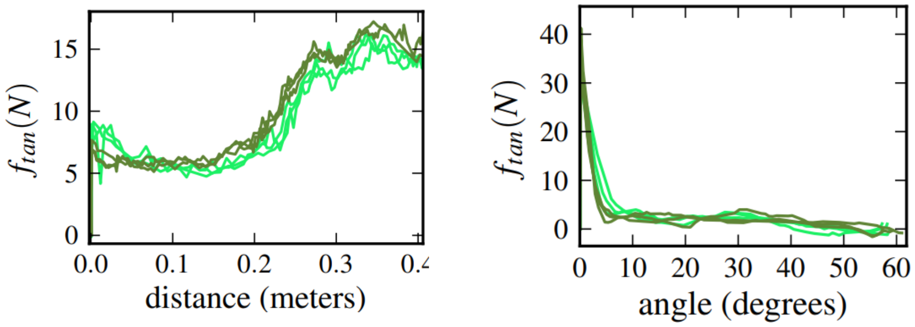
Even for some of the simplest articulated objects, there are different states the object can be in, which distinctly change the underlying kinematic model. This makes modeling these states important for autonomous interaction. A great example is a door. For a door, there are basically two states with respective articulation models, latched and unlatched. In the latched state, the door is closed, and there is a rigid relationship between the door and its frame. However, as soon as the door is unlatched (by articulating the handle), the articulation model changes, as a rotary relationship now connects the door and the frame. Detecting and describing these configuration-dependent changes in kinematic models is still subject to recent research. In work conducted by Kulick et al. [KOT15], pre-defined kinematic models are used to find ”joint dependencies”. Joint dependencies describe the logical relationship of different joints to each other. An example of a frequent relationship would be that one joint can lock/unlock one or multiple other joints. The presented framework builds on the insight that most mechanisms we interact with are designed for humans. Therefore, changes in the object’s joint kinematics are often marked by distinct haptic or visual feedback. These discontinuities in, for example, the forces encountered when articulating a joint can be captured. The joint configurations associated with the change points can then be used to segment the configuration space (e.g. range of possible articulations) into distinct segments in which the kinematic model stays the same and between which a change in the articulation model occurs. Change points are detected using Bayesian change point detection on joint friction time series data, estimated from FT sensor measurements. The robot is equipped with a kinematic model of the articulated object it interacts with and a controller that enables the robot to set the articulated object into a target joint configuration. One-step cross-entropy maximization is used in every step to find the target configuration of all joints of the articulated object that maximizes the change in the joint dependency distribution. Then the robot sets the object to the target configuration, using the known model. In the end, this yields a probability distribution that indicates whether the current joint is likely to be dependent on one of the other joints. Whereas this research is concerned with one-to-one dependencies between different joints and only assumes either no relationship or a locking/unlocking relationship, work presented by Niekum and Jain et al. [NOAB15, JN19] focuses on finding changes in the kinematic model of a single joint based on internal states. The first paper [NOAB15] presents CHAMP, an algorithm for online Bayesian change point detection. By employing the model formulation presented by Sturm and combining it with a probabilistic model for efficient change point detection, they show that their algorithm can robustly recover joint models and configuration-dependent joint model changes from 6-DoF rigid body pose data. The second paper [JN19] presents an extension to CHAMP, called Act-CHAMP and combine it with an algorithm to construct hybrid automata [LST12] from the identified models. In contrast to CHAMP, Act-CHAMP not only considers the visual feedback from object movement but also utilizes knowledge about the intended actions to detect change points in the joint configuration. In the presented work, a two-armed robot is used, where one arm ghosts the movements of the other. This allows a human operator to instruct the robot to articulate a mechanism with one arm, by moving the other arm. This way, intended actions can be recovered from the movements of the arm directly operated by the human. Knowledge about the actions taken on a joint allows, for example, to reason why object movement stopped, as this could either be caused by some model change (e.g. locking mechanism) or simply because the operator does not act on the object. Using the detected change points, their framework can then construct hybrid automata. These can be represented as a kinematic graph with edges that can change their joint type, dependent on the joint configuration. A clear limit of both of these approaches is that they currently only work on mechanisms with a single joint.

The approach presented in this work employs a model description similar to the one proposed by Sturm [Stu11]; however, the user should be more involved in the model identification such that simpler techniques can be used for model fitting.

2.2 Identification of Dynamic Models

As shown by Endres et al. in [ETB13], high-level information about the dynamics of an articulated object can be used to simplify the control and motion planning required for a manipulation task. Information about the forces needed to articulate an object can be used to interact with the object dynamically, as the robot has a model for how the object behaves even at higher speeds and accelerations. Furthermore, certain discontinuities in the object's behavior (like high opening forces) can quickly be estimated, rather than having to completely rely on control to, e.g. find the force at which movement starts.

There is a broad range of literature on automatic model identification for robot manipulators [HAJ06]. The approaches presented by these works are highly accurate. However, they also assume the availability of highly accurate, expert specified, precomputed models and very accurate, controlled measurements of each individual joint. This makes these approaches unpractical for our application. There are approaches that use simple dynamic models to identify parameters. These simplified models may not perfectly describe the real physical relationship but can be learned more easily and are powerful for making predictions about dynamic behavior. These include parameters like the threshold force needed to articulate an object, the generalized friction of a joint experienced during manipulation, and generalized mass or inertia parameters. Jain et al. [JNR$^+$10] present an approach for estimating the opening forces of doors and drawers with simple means. The presented method employs the kinematics fitting from Sturm [Stu11], to find the kinematics of doors and drawers from marker-based motion capture. Learned models are used to associate the encountered articulation forces with the joint articulation. They construct a gripper consisting of a 3D printed hook and an FT sensor. Then, the movement of all rigid bodies of the articulated object and the gripper, as well as the forces/torques encountered by the FT sensor, are recorded. For simplicity, it is assumed that no



(a) Tangential force profile of a drawer with telescoping rail

(b) Tangential force profile of a fridge door

Figure 2.3: Tangential force profiles of different mechanisms as recorded by Jain et al. in [JNR+10]

slipping occurs between the hook and a given handle and that the orientation of the hook relative to gravity stays the same throughout the measurement. Movement velocity is monitored so that quasi-static dynamics can be assumed. They find that even with the aforementioned simplifications, the measured data still has good accuracy and, as can be seen in Fig. 2.3, can show distinct dynamic properties of different mechanisms/objects. These experiments illustrate that one can find distinctive attributes of the dynamics of an articulated object using FT sensor measurements and motion capture.

A more sophisticated approach for approximating door dynamics is presented by Endres et al. [ETB13]. They present two main approaches for identifying the dynamic parameters of a door, 1) through vision-based observation or 2) through active articulation using FT measurements. They start with a simplified model that describes the door's opening angle θ over time, considering the starting angle θ_0, initial angular velocity ω_0, and constant friction α

$$\theta(t) = \frac{1}{2}\alpha t^2 + \omega_0 t + \theta_0 \tag{2.7}$$

which can be written as a least-squares problem

$$\theta_i = c_1 t_i^2 + c_2 t_i + c_3 = \mathbf{c}^T \mathbf{t}_i \tag{2.8}$$

where the parameter c_1 represents the friction coefficient. However, they find that a door's friction cannot sufficiently be described by a single friction constant, which Jain et al. [JNR+10] have already shown for the quasi-static case. Therefore, an opening angle dependent friction profile $\alpha(\theta)$ is trained using weighted least squares. A laser depth scanner provides training data for both the kinematic and the dynamic model. The door is swung open by a human operator, and the door's position is tracked. Using the kinematic model, the door's opening angle can be computed at every time step. What they find is that even the angle-dependent friction profile can not fully capture the door's movement. With further experimentation, they find that the dynamics apparently can not be described by a parametric model. Furthermore, the dependency on velocity also seems non-negligible. Therefore, a parameter-free Gaussian process [RW05] model is employed to learn a function dependent on articulation angle and angular velocity. For the second modeling approach, the robot should autonomously open the door, first with a constant velocity and then with an accelerated movement, to identify (constant) frictional torque and moment of inertia from FT sensor measurements at its wrist. Through experiments on different doors, it is shown that the Gaussian process model can accurately describe the door's movement. However, unsurprisingly given the previous results, the (simpler) model, trained from just the FT measurements alone, can only find a very rough approximation of the door dynamics. It is, therefore, only used as a first approximation step such that the robot can subsequently autonomously articulate the door to gather data for training the more complex model. We can see that although there is some promise in using FT sensor measurements to identify dynamic attributes of articulated objects, there is no conclusive research on how to use this data combined with gripper pose data to find useful models of object dynamics.

2.3 Use of Articulation Models in Robotic Manipulation

At this point, I would like to shed some light on different approaches that leverage generalized models of articulated objects in robotic manipulation tasks. Most of these approaches are either based on random sampling of end-effector poses, optimization of a given cost function, or control. In some cases, control-based methods can also enable manipulation without explicit knowledge of the articulation model.

In work presented by Burget et al. [BHB13], a random sampling-based approach is used to manipulate single joint articulated objects with a Nao humanoid robot. Kinematic models of articulated objects, which specify the joint type, movement, position, and handle position, are given. The planner is based on bidirectional tree search. Starting from the initial configuration and the goal state, configurations are randomly sampled from a precomputed set of stable robot configurations. By precomputing stable poses, only the constraints for the motion of the robot's 5-DoF arm have to be considered, and not the added stability constraints for the robot body. As most body poses are unstable configurations, this would increase the search space and slow down the algorithm significantly.

A straightforward control-based approach is the aforementioned method by Sturm et al. [SJS$^+$10], which uses equilibrium point control, introduced by Jain et al. in [JK09]. Equilibrium point control tries to follow a trajectory dictated by so-called equilibrium points, which specify the position the robot end effector would settle in, in the absence of externally applied forces. After specifying an initial grip location and pulling direction, in each step, the articulation model is queried for a Jacobian that indicates the direction in which the mechanism currently moves. Using this information and an appropriately chosen constraint vector to keep a steady grip, a new equilibrium point can be computed. With inverse kinematics, corresponding equilibrium angles for the robot joints can be determined and fed to the arm controllers. It is also shown that, assuming some uncertainty in the measurements, there is often sufficient similarity in different articulated objects to allow the robot to reuse and refine a previously learned model with data from another mechanism. In subsequent work [RSP$^+$12], the found models are also used to reconstruct the rigid bodies of the mechanism during articulation from camera footage and control the manipulation with position control.

An optimization-based approach is presented by Röfer [RBB20], who applies methods from optimal control to manipulate generalized articulation models. The problem is formulated as an optimization with a target state and several task constraints. A model-based configuration tracker is implemented to compute the configuration of objects in the scene by minimizing the difference between the perceived object pose and the underlying model-computed object pose. Using this configuration tracker and a formulation of the goal as a joint configuration, it is demonstrated that a robot in a simulated kitchen environment can keep the kitchen in the requested (orderly) state.

These examples illustrate that there are already approaches that can use models of articulated objects for autonomous manipulation. Although these methods still lack robustness and are mostly only designed to deal with simple mechanisms consisting of only one joint, this still verifies that kinematic object models which store information about joints and connectivity are useful for planning interactions with articulated objects.

Chapter 3

Operator guided Identification of Articulated Objects

As described in the introduction, the goal is to combine human intuition, which is of qualitative nature, and the precision of machines in one approach for identifying articulated objects. An overview of the proposed system can be seen in Fig. 3.1, it consists of the following parts:

- A gripper, used by the human operator to collect data on the interaction with the articulated object
- Interface to read out sensor data and prepare it for use in model identification
- Interface for the operator to directly input qualitative insights, monitor system performance, get instructions
- Back end that navigates through the series of measurements computes the single joint model fits and builds the overall model of the articulated object

Simulation should be used to build a first system implementation so that validation and testing of different approaches are possible before moving to the actual hardware. The following sections describe the different system components, what is investigated and implemented.

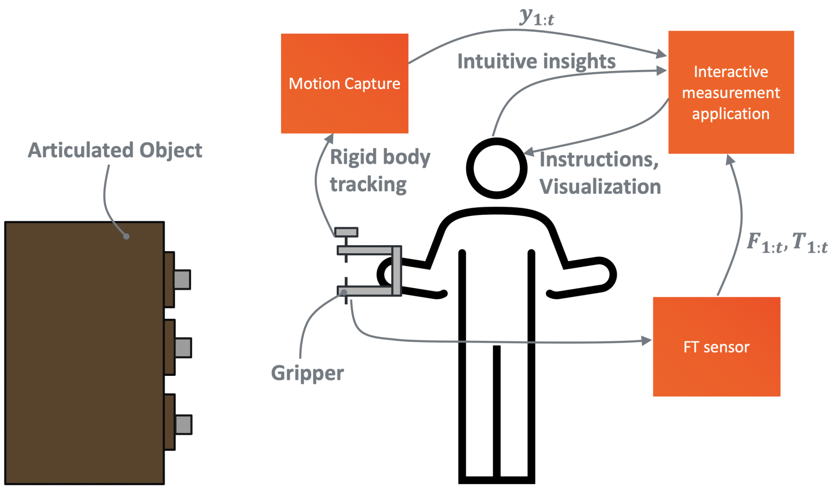


Figure 3.1: Overview of the proposed system

3.1 Measurement Hardware

3.1.1 Sensory System and User Interface

It is important to enable the operator to conduct reliable data acquisition easily. The inherent strength of having a human, not a robot, do the articulation steps is that a human will easily interact with the objects and use them in a natural way. However, in the end, the acquired data should allow a robot to articulate these objects. Therefore, the idea is to let the human "imitate" a robot by using a handheld device representing a gripper of which the position and dynamic state should be recorded. One should be able to rigidly contact a link of an articulated object, such that the joints can be articulated one at a time without the gripper slipping. Ideally, this device should be minimally obstructive to the operator manipulating the object. However, as this thesis consists of many more parts than just hardware design, in the actual implementation, we focus on designing and building a functional proof of concept system, which could then be improved upon in future works. The human operator should specify when measurement data is recorded, such that simple heuristics can be used to filter out the relevant from the irrelevant data. The user should be guided through the measurement process by the application, being prompted whenever a measurement is needed, or some other information should be specified. For the purposes of this thesis, this interaction can be implemented with prompts in the command line and visualizations of the measurement data and generated models through native python frameworks like *matplotlib* [Hun07]. This way, the operator can monitor in every step whether what was computed roughly fits the actual object. This is especially useful for avoiding simple errors, like wrong calibration of the measurement system, that would otherwise lead to confusion and lost time from nonsensical measurements.

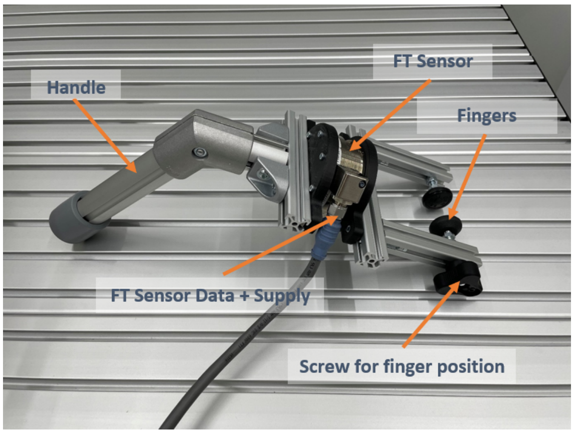


Figure 3.2: Self-constructed gripper with FT sensor and handle

3.1.2 Handheld Device

The hardware for collecting the articulation data should allow gripping and manipulating different parts of an articulated object, attaching an FT sensor and multiple motion capture markers. The focus of this work is not on the use of a special gripper but rather on collecting the movements of articulated objects throughout manipulation. An obvious choice for the gripper would be a standard electrical robot gripper. Two electrical gripper models were available for my project, the Weiss Robotics WSG 50-110 Fig. 3.3(a) and the Robotiq 2F-85 Fig. 3.3(b). However, by using a standard robot gripper, one would need an extra interface for controlling the gripper and a custom control strategy for grasping different size and shape objects. Furthermore, there would be limited ability to quickly adapt the fingers or size of objects that can be grasped. As a human has to operate the device, most of the time with a stretched-out arm, some consideration should be given to the weight of the gripper. Most robot grippers are relatively heavy. The two models available to me weigh 1.2 kg for the Weiss Robotics gripper and 0.9 kg for the Robotiq gripper. Both would still need an attachment for the FT sensor, the FT sensor itself, and some handle for the human to grasp. Another consideration is the amount of cabling required to get the gripper to work, as it should be minimized for ease of use and also lowered weight. For both considered grippers, at least two lines would be needed, one for the communication and the other for supplying the gripper.

For these reasons, a custom gripper is constructed (seen in Fig. 3.2) that can be closed manually and allows easily attaching an FT sensor, different ”fingers”, and a handle for the human operator to grip. To get to a viable solution as quickly as possible, this gripper is constructed from standard ”Item Industrietechnik” profiles that can be modularly put together to get the desired shape. Grasping is realized as a two jaw mechanism. With the addition of two 3D printed plates attached to two screws, one on either side, the gripping mechanism works like a vice. One side is rigid, and on the other one, the screw can be screwed in or out to tighten or loosen the clamp. Next to general joints also handles should be identified. Therefore, the feet are printed with a V-shaped groove so that there is better guidance when rotating around- and moving along a handle. The FT sensor can be attached through a 3D printed custom mounting plate attached to the item profiles. The handle consists of a 45° angulated tube, directly attached to the FT sensor.

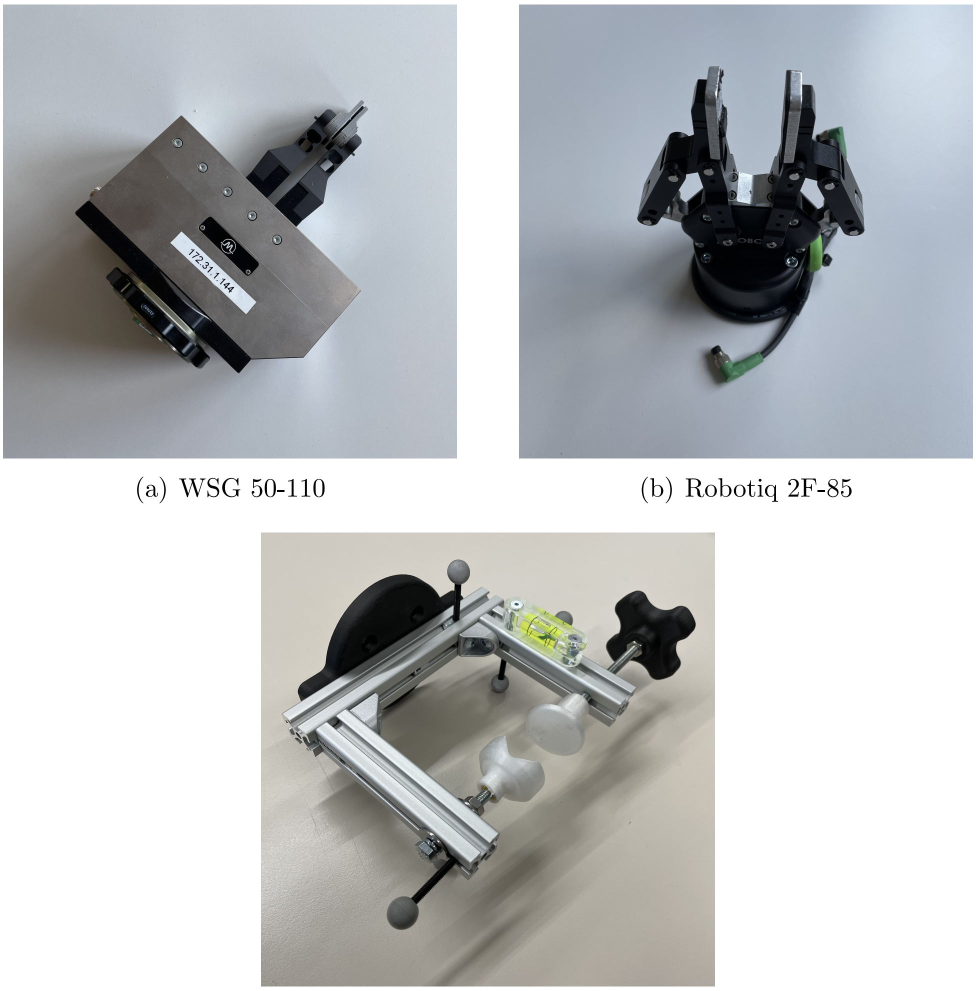
The gripper was tested by attaching it to different objects and moving them. With enough clamping force, even heavy to articulate mechanisms can be articulated. However, there is a problem when trying to open mechanisms that need a very high initial force. Additionally, when loosening the grip, due to the applied force, sometimes the foot unscrews from the screw so that it has to be reattached. The second issue could be mitigated by exchanging the mobile foot by a foot with a ball joint. It also has the added benefit that by enabling the foot to change orientation with respect to the object being grasped, the grip is a lot tighter.

3.2 Model Identification

3.2.1 Model Formulation

The main information needed for interacting and manipulating an articulated object is its kinematic structure. The information gathered on this structure should allow querying the model for positions of the different rigid bodies. This, in the end, should enable an external planning algorithm/controller to generate an articulation trajectory for bringing the object into a requested joint configuration/state. However, planning articulation tasks is a much higher level task that is not inherently coupled with the articulated object. Therefore, it should not be investigated further throughout this thesis.

The structure and movement of any articulated object are mainly determined by its joints and their connectivity structure. The models should thus be described as a collection of joints interconnected through straight links. As most articulated objects in everyday life are kinematic trees, the formulation and fitting of these models should focus on kinematic trees, with no special consideration of kinematic loops. Articulated objects intended to be used by humans normally present some kind of handle with the sole purpose of allowing a person to articulate the mechanism easily. This assumption could be used as prior information for planning, which is why dedicated handle models should be included in the identified kinematic models. This way, the model could provide a gripping pose associated with the handle so that querying for a gripper trajectory is possible. Burget et al. use a similar formulation in [BHB13], to autonomously open simple articulated objects with a humanoid robot. In general, there are many parts on an articulated object that one could use to grip and manipulate it. However, fully modeling these would require



(a) WSG 50-110

(b) Robotiq 2F-85

(c) Item Gripper

Figure 3.3: Considered Gripper Models

building a fully volumetric 3D model, which significantly increases the complexity of model identification and planning for robotic interaction. Learning a handle model allows a much simpler definition of a grasping pose useful for interaction with most articulated objects.

With a kinematic tree of all included joints and the handle, the model can already capture the effects of manipulations on the object. Additionally, the model can store information on the joint limits of every joint. Joint limits are valuable information for articulating the object, as they pose (hard or soft) constraints on the range in which every joint can be articulated. When the joint limits are not known, this could lead to a robot running into them during manipulation, either being confused about what happened or breaking something. It is very intuitive for a human interacting with these objects where the limits are, so this information can easily be gathered by having the operator mark the limits. Joint dependencies and hybrid joint models, as mentioned in Sec. 2.1, are not investigated in this thesis. Joint dependencies could be tracked through an extra user prompt for specifying logical dependencies between the joints after the overall kinematic structure is specified. Hybrid joint models can be learned with one extra measurement for every extra joint model of a hybrid joint.

As already mentioned in the introduction, including dynamic information in the model could allow more sophisticated manipulation and simplified control. However, there is also plenty of ambiguity in these parameters and how they should be represented in a model. The interface and measurement of the force-torque sensor are implemented, and some preliminary tests are conducted. However, as robust identification of kinematic models is needed first to identify dynamic attributes, identifying dynamic models remains a topic for potential consecutive works.

3.2.2 Parametric Joint Models

Most of the articulated objects encountered in everyday environments consist of rotary and prismatic joints. Most other mechanisms can be described as a combination of these two. Furthermore, Sturm's work [Stu11] already presents a sophisticated framework that employs parameter-free joint models when the standard ones do not work. Therefore, as the focus of this work is to get to a minimum viable system, only pre-defined parametric models are considered. To make it possible to add in additional joint models in the future, each joint model is implemented as its own class that stores joint-specific parameters and inherits from a parent joint class that implements all general attributes associated with a joint. This way, a new joint can be added by adding an additional class that implements joint-specific parameters, like the forward and inverse kinematic function, and an additional function for the fitting procedure.

The following section presents the model formulations used. The joint formulations are adopted from the formulations presented by Sturm in [Stu11]. To describe the kinematic relation between two object parts, the operators $\oplus$ and $\ominus$ are used, where $\oplus$ describes the motion composition and $\ominus$ its inverse. In the case that $x_1, x_2 \in \mathbb{R}^{4\times4}$ represent homogeneous matrices, the $\oplus$ operator corresponds to matrix multiplication $x_1 \oplus x_2 = x_1 x_2$ and the $\ominus$ operator corresponds to the inverse multiplication $x_1 \ominus x_2 = (x_1^{-1})x_2$. In the following models, the output z of the forward kinematics functions is a frame matrix $z \in \mathbb{R}^{4\times4}$, the input z to the inverse kinematics functions is simply a point in 3D space without an attached pose $z \in \mathbb{R}^{3\times1}$, as this represents the considered input from the motion capture system.

The movement of a ***Prismatic*** joint can be represented by a line in 3D space. This line has two limits associated with it, which represent the lower and upper joint limits. The forward kinematic function of a prismatic joint can be represented by (3.1). Here $a \in \mathbb{R}^{4\times4}$ is the center frame of the prismatic joint, which is positioned at one of the joint limits subsequently associated with $q = 0$. The vector $e \in \mathbb{R}^{3\times1}$ represents the unit direction vector of the joint axis in the joint's coordinate frame. For simplicity, the frame a will always be oriented such that e is aligned with the frames z-axis, so $e = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$.

$$\mathbf{z} = f(q) = \mathbf{a} \oplus (\mathbf{e} \cdot q) \tag{3.1}$$

$$q = f^{-1}(\mathbf{z}) = \mathbf{e^T} \mathit{trans}(\mathbf{a} \ominus \mathbf{z}) \tag{3.2}$$

Here $\mathit{trans}(\cdot)$ extracts all translational components of a frame.

A ***Rotary*** joint can be represented by a circle in 3D space. As with the prismatic joint, there are two limits associated with this circle. The forward kinematic function of a rotary joint can be represented by (3.3). $c \in \mathbb{R}^{4\times4}$ is the center frame of the rotary joint, which is positioned at the calculated circle center. $r \in \mathbb{R}^{4\times4}$ represents a child frame attached to the rotary joint. Again, for simplicity, the z-axis of the frame c is chosen as the axis of rotation. The x-axis is set to mark one of the joint's limits, defined as $q = 0$.

$$\mathbf{z} = f(q) = \mathbf{c} \oplus \mathit{Rot}_z(q) \oplus \mathbf{r} \tag{3.3}$$

$$q = f^{-1}(\mathbf{z}) = \mathit{Rot}_z{}^{-1}(c \ominus \mathbf{z}) \tag{3.4}$$

Here $\mathit{Rot}_z(\cdot)$ conducts a rotation around the z-axis by q.

3.2.3 Parametric Handle Models

The handle is intended to be the part of the model that is directly grasped during manipulation. Therefore, it should provide a frame that contains information about the angle and position at which the handle can be gripped. The presented models cover the main categories of handles encountered on everyday objects. In line with work by [MMB17], the system considers three handle models: 0-DoF handle, 1-DoF handle, and 2-DoF Handle. The degrees of freedom describe how many degrees of freedom there are between the gripper and the handle. For these handle models, the input-output of the forward and input of the inverse kinematics function z is a frame matrix $z \in \mathbb{R}^{4\times4}$.

A ***0-DoF*** handle can be represented by a single frame $a \in \mathbb{R}^{4\times4}$ that is rigid in its position and orientation. The frame a is identified by taking the average over a sequence of recorded gripper frames. An inverse kinematics function is not needed, no joint articulation is possible.

$$\mathbf{z} = f(q) = \mathbf{a} \tag{3.5}$$

The ***1-DoF*** handle has one rotational degree of freedom, so the gripper can change the orientation in which it attaches to the handle along a rotational axis. Therefore, the 1-DoF handle can be described as a rotary joint with a center frame that rotates about its z-axis. The singular difference is that in this model, there is no child frame attached. Therefore, the only change that occurs is in the orientation of the center frame in the WCS. A user-specified radius is stored with the model.

$$\mathbf{z} = f(q) = \mathbf{c} \oplus Rot_z(q) \tag{3.6}$$

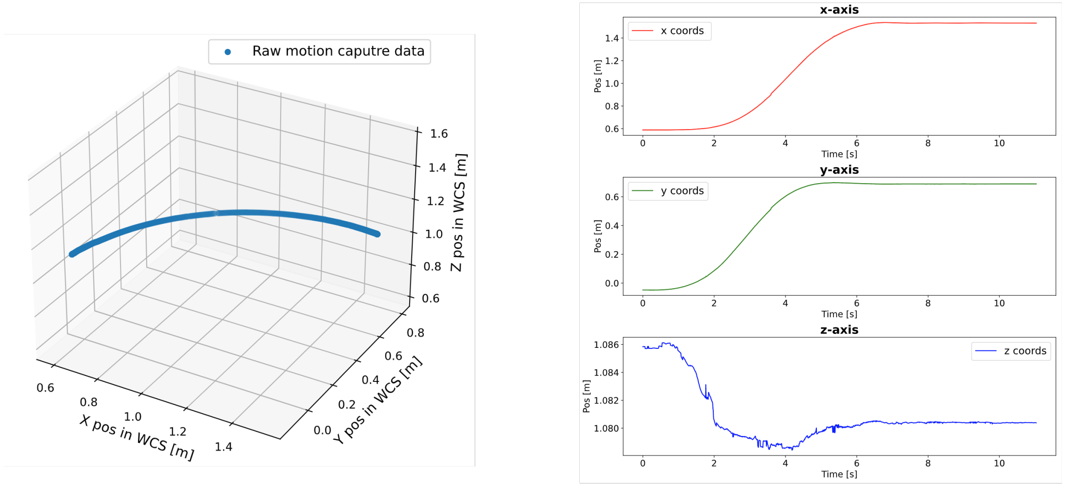
$$q = f^{-1}(\mathbf{z}) = {Rot_z}^{-1}(\mathbf{c} \ominus \mathbf{z}) \tag{3.7}$$

The ***2-DoF*** handle allows rotation around and translation along a central axis, therefore, we combine the prismatic and the rotary model. As it consists of two models, the model has four limits, two for the rotary component and two for the translational component. As with the other two handle models, the model does not describe a point on its surface but rather a frame positioned on its central axis. The z-axis of the frame coincides with the handle's central axis to represent the axis of rotation and translation. Like for the 1-DoF handle, the user specifies a radius during model identification.

$$\mathbf{z} = f(\mathbf{q}) = (\mathbf{a} \oplus \mathbf{e} \cdot q_1) \oplus Rot_z(q_0) \tag{3.8}$$

$$q_0 = f_0^{-1}(\mathbf{z}) = {Rot_z}^{-1}(\mathbf{a} \ominus \mathbf{z}) \tag{3.9}$$

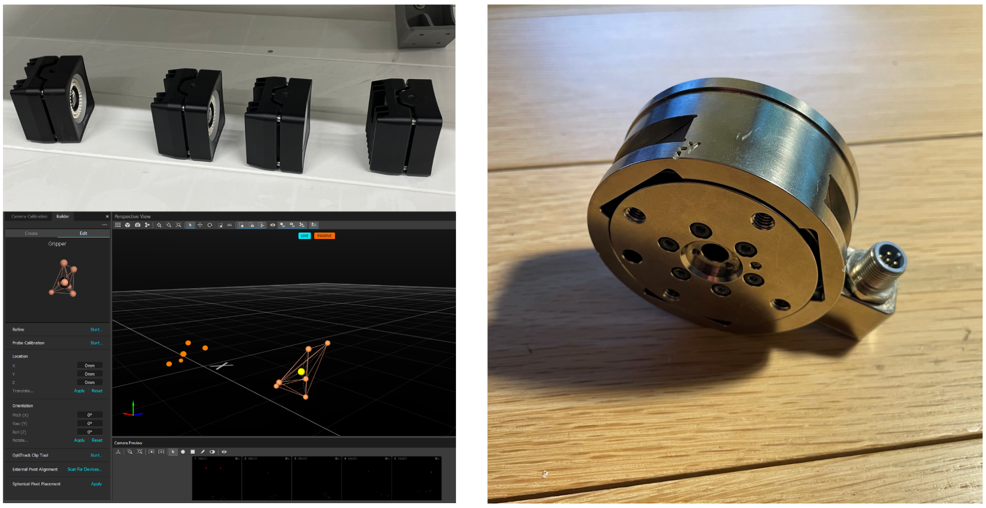
$$q_1 = f_1^{-1}(\mathbf{z}) = \mathbf{e}^{\mathbf{T}} trans(\mathbf{a} \ominus \mathbf{z}) \tag{3.10}$$



(a) 3D Motion Capture measurements of articulating a rotary mechanism

(b) Same Motion Capture measurements visualized for each axis seperately

Figure 3.4: Sample position data from the motion capture system



(a) OptiTrack cameras and rigid body tracked in Motive

(b) ATI Gamma FT sensor

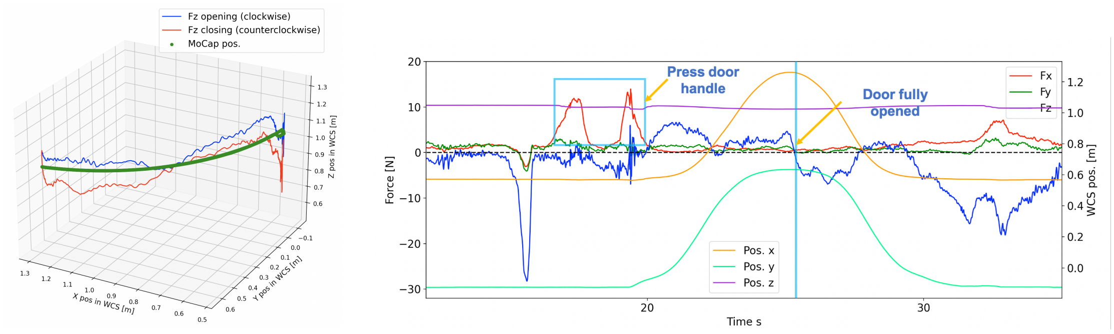
Figure 3.5: Used sensors

3.2.4 Data Acquisition

Initially, the use of two sensors is considered, a marker-based motion capture system, seen in Fig. 3.5(a), primarily to gather movement data for the kinematic identification, and an FT sensor, seen in Fig. 3.5(b), to gather information for identifying dynamic attributes. The measurement application needs to interface to both sensors, read them in real-time, align the data streams in the time dimension, and maybe do some basic filtering. As the sensory hardware used is of high fidelity, strong measurement noise is not expected. However, the measurement procedure and the hardware that the sensory equipment is attached to could introduce non-negligible errors that have to be coped with.

In tests with the actual motion tracking hardware, it is found that a capture rate of 120Hz is sufficient for the fitting to work reliably. In comparison to the movements, the sampling is so quick that, if no occlusion occurs, the position samples are only millimeters apart, leading to smooth trajectories as seen in Fig. 3.2.3. Furthermore, the data sent by the motion capture system is very accurate and has minimal noise. Throughout the experiments, for a standard calibration in a prepared workspace (camera setup, marker placement, minimized amount of reflections), the estimated positional tracking error $\epsilon_{y,pos}$ is never higher than 0.8 mm (according to calibration results from the tracking software). The nominal noise of the system is $\sigma_{y,pos} < 0.0005$m and $\sigma_{y,orient} < 0.001°$. There are rarely outliers due to tracking errors. The tracking software recognizes when the positional accuracy of a marker decreases. In case of deviations beyond a threshold, it stops tracking and continuously sends the last confidently tracked point. Therefore, a problem in the tracking data can be identified by monitoring the tracking data for sequences of constant values. The threshold for whether the rigid body tracking went out of scope can be adjusted so that one can choose a good trade-off between tracking accuracy and continuous tracking data. To combat duplicates in the tracking data due to a rigid body running out of scope, a simple check is implemented that removes consecutive measurements that are identical. As long as there is still a sufficient part of the trajectory where no tracking error occurred, removing this data does not seem to have a negative effect. Interpolating sections where no data is available, using previous measurements or other sensor data, could potentially improve accuracy and stability, but as no problems were encountered in testing, this is not considered further.

Tests with the FT sensor show promise that it is possible to pick out some of the key points in the trajectory of a mechanism from the data. An example is given in Fig. 3.6, where a door is opened using the gripper with the attached FT sensor and motion tracking. In Fig. 3.6(b), some key points are marked. The door handle is pressed two times, which is indicated by distinct spikes in the force profile of Fx. The peaks represent when the joint limit is reached. After the door is unlatched, the door hinge is actuated, which can be seen in Fz's force profile and the positions measured by the motion capture system. As the movement is initiated, a relatively high Fz is needed to start the door's movement. When the door starts moving, the force quickly decreases in reaction to the movement and picks up again so that the door moves with a roughly constant speed. When the door limit is reached, the force reverses to stop and reverse the movement. The associated 3D plot in Fig. 3.6(a) shows the opening movement with the associated force Fz. Although there are clearly identifiable trends in this measurement, it can also be seen that there is a strong variance in the measured force. Throughout this measurement, the gripper orientation with respect to the door is held constant so that it is possible to bias the FT sensor at the beginning of the measurement with the measured gravity force. The gripper hardware itself, combined with the FT sensor, weighs around 500 g. To evaluate the effect reorienting the gripper has on the FT measurement, the force profile in different gripper poses is measured, a sample result of these measurements can be seen in Fig. 3.7. It can be seen that the force of gravity already contributes to a very noticeable force acting on the FT sensor when compared to the forces encountered in the previous measurement of the door. This effect could be mitigated through standard gravity compensation [YSL21], commonly implemented in robotics applications with FT sensors to remove the gravity-induced distortions on FT sensor measurements. The tests show the promise of using an FT sensor in a handheld device to identify articulated objects' dynamic attributes. In the next step, a measurement procedure would have to be devised that is robust against disturbances introduced by the human operator and manages to generalize the process over a broad range of joints and mechanisms. However, before the data can be used to identify model dynamics, robust identification of kinematic models is needed, which is the focus of this work. Therefore, these first experiments should only show



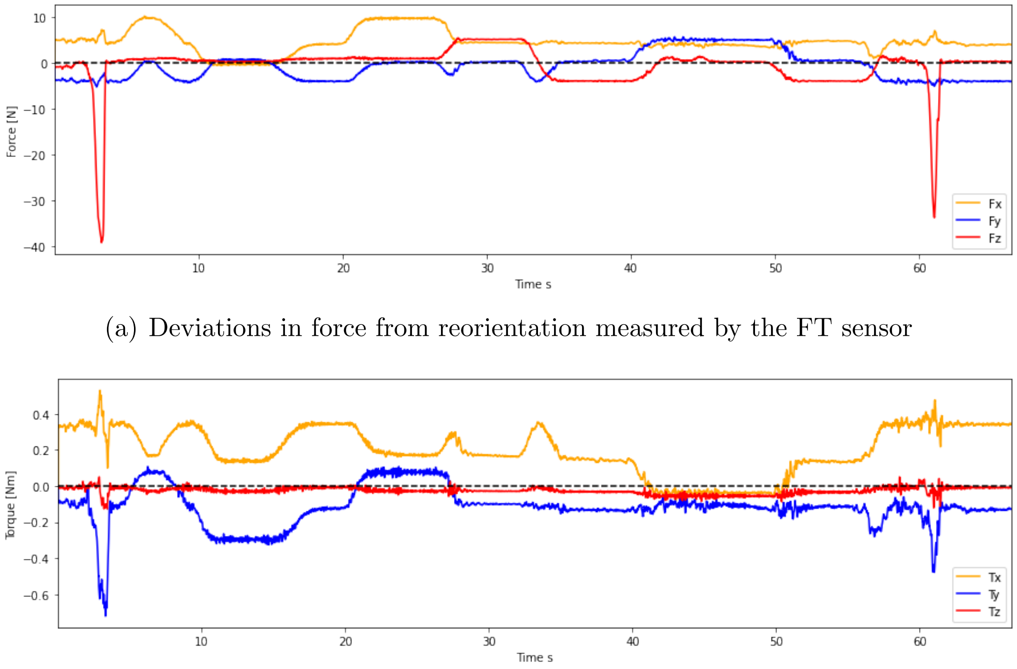
(a) Motion capture data overlayed with Fz

(b) Motion capture data overlayed with FT measurements

Figure 3.6: Motion capture and FT sensor measurements during door opening

that there is promise in further investigating integration of the FT sensor for identifying dynamic attributes of articulated objects.

Both the motion capture system and the FT sensor connect to the fitting application via UDP. An ethernet switch is used to bundle all the ethernet connections so that only one line has to be connected to the application PC. The FT sensor data can directly be decoded using the specification of the frame format published by ATI. The raw data from the motion capture system could also directly be read from the UDP frames. However, this data is of little use without a considerable amount of processing. Therefore, the proprietary software Motive, published by the company that produces the motion capture system, is used. The application has an uncomplicated interface and allows to set up a capture scene quickly. At the start of every capture session, the system must first be calibrated, which can be done directly through Motive. Afterwards, the objects that should be tracked can be marked in Motive, and the resolved pose data can be streamed to the measurement application via UDP loopback to be read and unpacked using the provided python client. For the purpose of this work, only the gripper has to be tracked. This can conveniently be done through so-called rigid body tracking, which allows specifying a set of markers that identify a rigid body in the scene. However, there are two inherent problems with rigid body tracking using Motive. First, the so-called pivot point (point on the object that is tracked) is positioned "randomly" (geometric center of the associated markers), which normally does not coincide with the tip of the gripper; second, upon creation, the orientation of the rigid body is set so that its frame is aligned with the WCS, so it is normally not aligned with the FT sensor frame. To combat these issues, a calibration step has to be included in the measurement application to identify the gripper tip and align the tracked frame with the FT sensor frame. Two approaches for each are implemented. The first simply involves bringing the gripper into a known position and orientation to compute the offset. However, as this is not always easily possible, as, in every new capture



(a) Deviations in force from reorientation measured by the FT sensor

(b) Deviations in torque from reorientation measured by the FT sensor

Figure 3.7: Gravity effect on FT measurements

scene, the WCS will be set differently, a second method is implemented, which takes advantage of the geometry of the gripper. When attached to a cylindrical handle, there are two major rotation axes associated with the gripper. One is specified by the screws used to clamp down on the handle. The other is specified by the skid in the gripper foot and the major handle axis. With two subsequent rotations of the gripper, these axes can be identified, their intersection (normally point of closest distance) can then be used as the gripper tip. As the FT sensor is aligned with the gripper axes, the identified axes can also be used to approximate the orientation of the FT sensor frame relative to the WCS. The accuracy of this procedure is tested in the evaluation.

3.2.5 Fitting Algorithm

The idea is to leverage human intuition throughout the model fitting such that the process is more transparent and the most useful insights of the operator can be included. For these reasons, rather than having the program find the model of a particular joint and the overall connectivity structure of the articulated object, the operator should specify both the joint types and their connectivity for a given articulated object. The operator can select from a set of models. Subsequently, recorded data is then used to find a fit for the specified model. As the operator specifies which type of joint is currently being identified, and each joint is based on a well-defined parametric model, least-squares based fitting methods are chosen.

A line in 3D coordinates can be parameterized by a support vector $\mathbf{v}$, a direction vector $\mathbf{r}$ and a scaling parameter t. Any point $\mathbf{p}$ on the line can be computed by scaling the direction vector with the parameter t.

$$\mathbf{p} = \mathbf{v} + t\mathbf{r} \tag{3.11}$$

As described in [GPL17], we can formulate a least-squares problem for finding a line that "best" fits through a set of data points by reorganizing this line description,

$$p_x = ap_z + b \tag{3.12}$$
$$p_y = cp_z + d \tag{3.13}$$

with

$$a = \frac{r_x}{r_z}, \qquad b = v_x - \frac{r_x}{r_z}v_z, \qquad c = \frac{r_y}{r_z}, \qquad d = v_y - \frac{r_y}{r_z}v_z \tag{3.14}$$

As (3.12) and (3.13) are both linear equations with two unknown parameters we can solve them using the standard linear least-squares solution to get an estimate for the parameters a, b, c, d. The direction vector of the line, $\mathbf{r}$ can now be computed as the cross-product of the two normal vectors of the planes described by (3.12), (3.13).

$$\mathbf{n}_1 = \begin{bmatrix} 1 \\ 0 \\ -a \end{bmatrix}, \qquad \mathbf{n}_2 = \begin{bmatrix} 0 \\ 1 \\ -c \end{bmatrix}, \qquad \mathbf{r} = \mathbf{n}_1 \times \mathbf{n}_2 = \begin{bmatrix} a \\ c \\ 1 \end{bmatrix} \tag{3.15}$$

As the least-squares fit of the straight line goes through the average of the data, the support vector can be estimated by computing the average of all data points.

$$\bar{v}_x = \frac{\sum_{i=1}^n p_x^i}{n}, \qquad \bar{v}_y = \frac{\sum_{i=1}^n p_y^i}{n}, \qquad \bar{v}_z = \frac{\sum_{i=1}^n p_z^i}{n}, \tag{3.16}$$

To fully define a circle in 3D coordinates, we need to specify a plane the circle resides in, a center point, and the circle radius. A point $\mathbf{p}$ on the circle can be computed from two orthogonal vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ defining the plane, the radius r and an angle θ.

$$\mathbf{p} = \mathbf{c} + \cos(\theta) r\mathbf{v}_1 + \sin(\theta) r\mathbf{v}_2 \tag{3.17}$$

The algorithm presented in [PCA16] is used to fit a circle to the measurement data. In this approach, at first the plane of the circle is identified using *principal component analysis* (PCA) [BK19]. As PCA identifies the directions of highest variance in the mean-normalized data, the normal vector of the circle can be approximated from the third principal direction (least variance). In the next step, the data can be projected onto the identified plane, such that it is reduced to two dimensions. In 2D, a circle is defined by the following equation

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \tag{3.18}$$

where $\begin{bmatrix} x_c & y_c \end{bmatrix}^T$ represents the circle center, r represents the circle radius, and all vectors $\begin{bmatrix} x & y \end{bmatrix}^T$ that fulfill the equation are points located on the circle. This representation can be rearranged to get a standard least-squares problem

$$c_0 x + c_1 y + c_2 = x^2 + y^2 \tag{3.19}$$

$$\mathbf{Ac} = \mathbf{b} \tag{3.20}$$

with

$$c_0 = 2x_c, \qquad c_1 = 2y_c, \qquad c_2 = r^2 - x_c^2 - y_c^2 \tag{3.21}$$

$$\mathbf{A} = \begin{bmatrix} x_0 & y_0 & 1 \\ . & . & . \\ . & . & . \\ x_{n-1} & y_{n-1} & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} x_0^2 + y_0^2 \\ . \\ . \\ x_{n-1}^2 + y_{n-1}^2 \end{bmatrix} \tag{3.22}$$

After identifying the central point and radius of the circle from the 2D data, the center can be transformed back into 3D coordinates. After adding the previously subtracted mean, we know the circle center, central axis, and radius, all parameters needed to fully define the circle.

Although the data gathered from the motion capture system is in most cases very clean and varied enough to reliably find accurate fits for these simple models, issues could still occur when the operator does not perform a clean movement. Therefore, robustness against outliers is added through a basic implementation of RANSAC [FB81]. In every iteration, the RANSAC algorithm computes a model fit on a minimal set of randomly sampled data points. Therefore, one should provide a simple fitting scheme that works on a small set of data samples such that the algorithm only samples a negligible amount of data points in each iteration and runs quickly. For the line model, the simplest fitting procedure is provided, which only requires sampling two points. One of these points is then used as the support vector, and the difference between the other point and the support vector is used as the direction vector of the line. For the circle fitting, the outlier detection is only conducted to remove outliers that do not lie on the circle's plane, as this is the part of the fitting that is most vulnerable to outliers due to PCA being prone to outliers [SV08]. Therefore, 3 points are sampled, and subsequently, the plane they reside on is computed by using one vector as the support vector and the others to compute the direction vectors. The error is then calculated by taking the Euclidean distance between a point and the respective line or plane.

3.2.6 Measurement Process

The proposed system should be the basis of an expandable toolkit for identifying and modeling attributes of articulated objects. User interaction is a key component in the flexibility and expandability of the approach. The human operator can quickly react to unknown prompts, perform novel articulations and assist the software when it is uncertain or runs into an unknown situation. Therefore, the software should provide real-time interaction with the user, presenting prompts whenever a new measurement should be conducted, or other information is needed. Live visualization of measurement data and fits is beneficial so that the user can quickly assess if something went wrong.

Figure 3.10 presents a flowchart of the implemented measurement application that illustrates the general procedure for measuring an articulated object. At the program start, the user can input joint types and connectivity of the articulated object. The next step implements the calibration procedure mentioned in Sec. 3.2.4, to identify the relative position of the gripper tip with respect to the currently tracked point. Afterwards, the program goes through the kinematic tree depth-first to identify the parameters for each joint or handle one by one. For each identification, in the first step, the user is asked to measure the (full) joint articulation so that the joint parameters can be fit to the trajectory. Next, the user is asked if the joint limits should be specified manually. If the user selects no, the program will simply use the start and endpoint of the trajectory as the joint limits. When the operator selects yes, the limits can be specified by two measurements. For each of these, the user has to first bring the joint into the joint limit, then the point is recorded, and the limit is computed by projecting the averaged measurement onto the computed joint, using the joint's inverse kinematic function to find the articulation. For the rotary joint, the frame is then set such that the x-axis runs through the lower joint limit, which subsequently is identified as $q = 0$. For the prismatic joint, the frame is placed at the position of the lower joint limit, such that the lower limit corresponds to $q = 0$; for the handles, this is done similarly. The process of joint fitting stops when all joints are identified. The measured data and joint fits are visualized for a last visual check, and the full model is stored for later use.

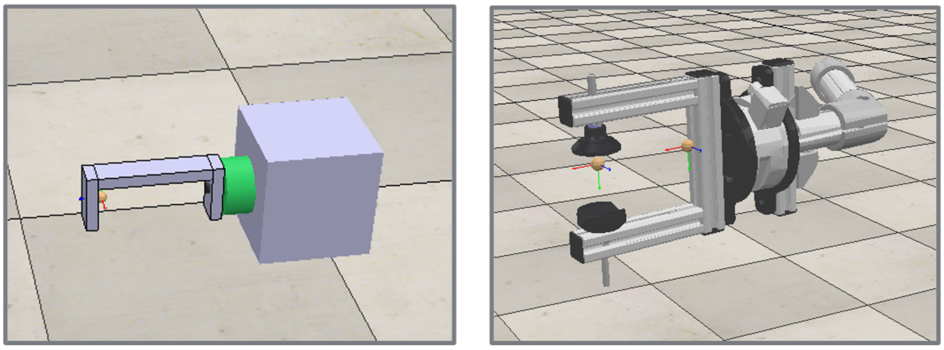
3.3 Use of Simulation for Testing the Model Identification

The use of simulation should speed up the process of implementing the system on actual hardware by already providing a method to validate the methodology before working extensively with the hardware. However, therefore it is also important that the simulation is reasonably close to the real world so that it can be assumed that performance, at least partly, translates to the hardware system. One important factor is that the user does not operate a mechanism along an ideal trajectory. The simulation should consider this. Furthermore, contact effects like slipping on a handle are likely to occur. A reasonable library of objects that adhere to their general working principle in the real world is constructed to conduct a broad range of tests. For a simplified transition to the actual hardware system, the general pipeline using simulation should be very similar to the pipeline used with the hardware. Therefore, already in the simulation step, the interfaces are chosen accordingly to make the transition easier.

For the simulations, I decided to use the CoppeliaSim simulator from Coppelia Robotics [RSF13]. CoppeliaSim Edu Version 4.2.0 for Windows is used. The simulator offers all major features needed for testing the system. It offers a GUI frontend for building the simulation scene from primitive or more complex objects. There is a mesh importer that allows importing external CAD data to subsequently model it for simulation. Every scene object can be equipped with a child script to run code during simulation. There is a direct interface to the Qt GUI framework [Com] to create simple GUI elements inside the simulator. The simulator offers kinematic as well as dynamic simulation with limited contact physics.

The constructed simulation environment can be seen in Fig. 3.9. To evaluate the full measurement procedure, real-time communication between CoppeliaSim and the measurement application is implemented. This way, the simulator can be started through the application, and measurements can successively be collected. A simple GUI is implemented that allows the user to articulate a mechanism with the gripper and also specify when a joint recording should start or stop. To get a realistic simulation, the actual gripper used in the real-world experiments is modeled in CAD and imported into the simulator. The model can be seen in Fig. 3.8(b). To get a realistic opening movement that considers the effects described above; the gripper is controlled by applying a force vector rather than specifying an opening trajectory for the gripper to follow.

CoppeliaSim offers multiple APIs to interface with the simulator through different programming languages. For this implementation, the B0-based remote API is used. It allows an external python application to fully control the simulator live. A python client that implements the underlying communication protocol and standard

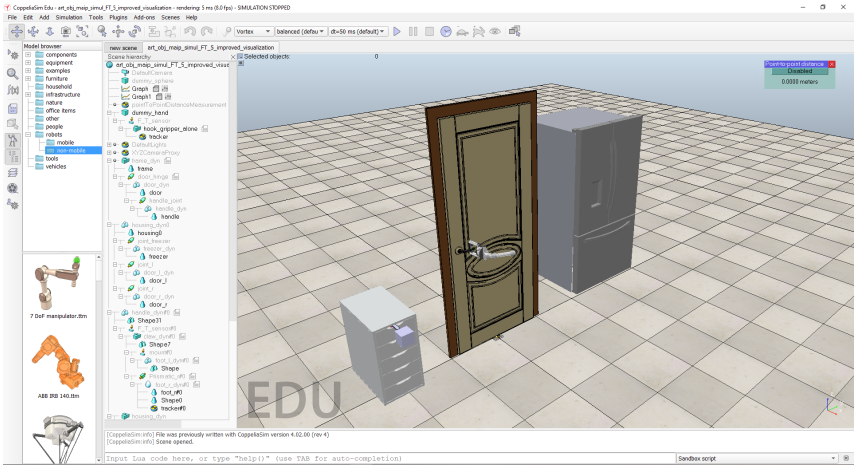


(a) Simplified gripper model

(b) Modeled gripper

Figure 3.8: Gripper models implemented in simulation

API functions is provided by CoppeliaSim. This way, the simulator can be directly controlled through the measurement application. To change from simulation to real-world measurements, only the client has to be changed. The rest of the software can already be tested and should run very similar to the tests with the simulator.



CoppeliaSim Edu - art_obj_maip_simul_FT_5_improved_visualization - rendering: 5 ms (8.0 fps) - SIMULATION STOPPED

Vortex | balanced (defau | dt=50 ms (default)

Selected objects: 0

```
[CoppeliaSim:info] File was previously written with CoppeliaSim version 4.02.00 (rev 4)
[CoppeliaSim:info] Scene opened.
```

Figure 3.9: Simulated articulated objects for evaluation

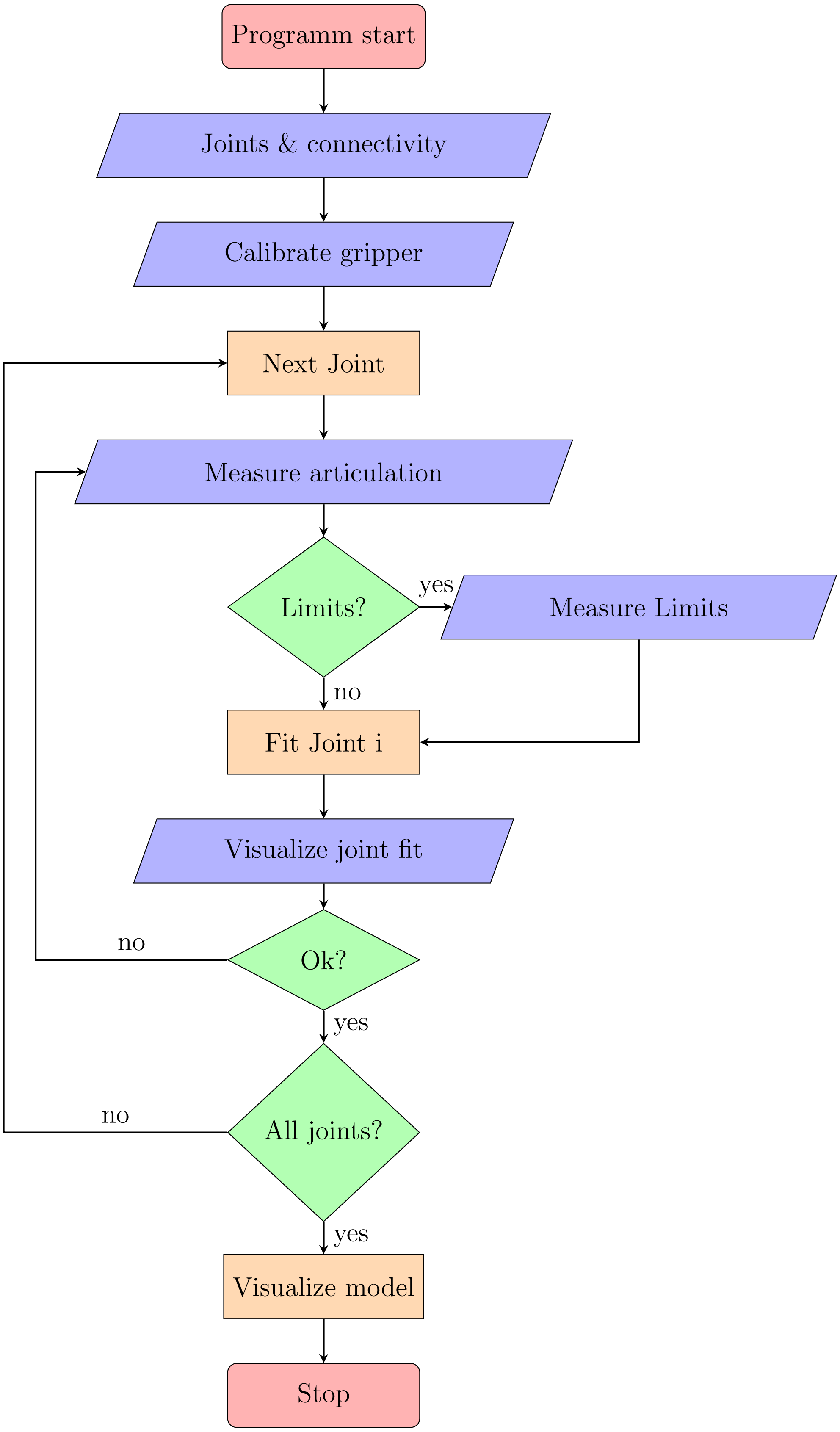


Figure 3.10: Program flow diagram for the implemented measurement application

Chapter 4

Evaluation

To evaluate whether accurately identifying the kinematic structure of articulated objects is possible with the previously outlined system, tests are conducted on the real-world hardware. The following sections give an outline of the conducted tests, present the results and assess how good the procedure works and what could or should be improved.

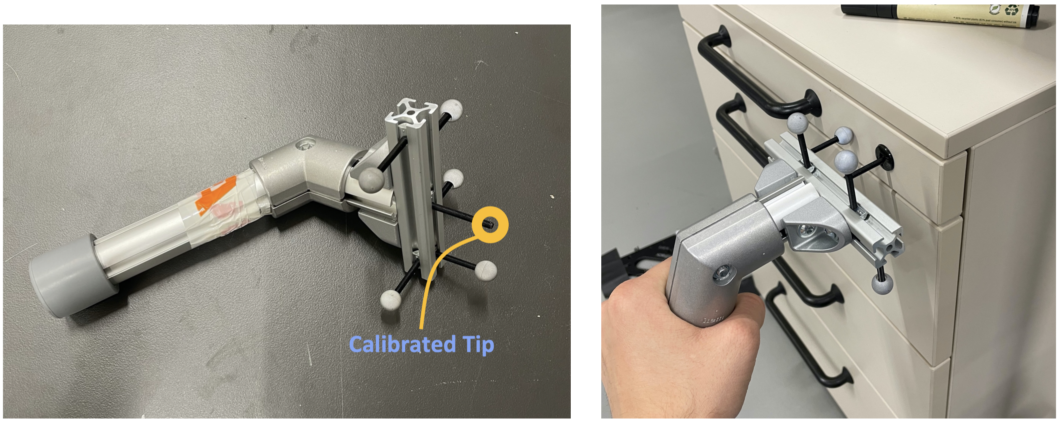
4.1 Setup

4.1.1 Measurement Hardware

The aforementioned self-built gripper is used in conjunction with the motion capture system optiTrack primeX13 [Nat]. The motion tracking is run at a frequency of 120 Hz. Reflective markers are attached to the gripper for the motion capture, as shown in Fig. 3.3(c) (a minimum set of 3 markers is required for tracking). The motion capture system is set up in a semi-circle around the articulated object. Four to five cameras, mounted on tripods at around 1.5m height, are placed to capture the entire articulated object and the WCS origin placed on the floor in the vicinity of the articulated object (Fig. 4.2(b)). To have consistency throughout the measurements, the WCS origin has to be placed at the same position such that the coordinate systems are as equal as possible. The gripper can then be tracked as a rigid body inside the Motive software, and all rigid body data is streamed via UDP loopback to the measurement application.

4.1.2 Metrics and Procedure

To assess the accuracy of the fit models, the fitting is tested on single joints. To get an approximation of the ground truth, axes are manually measured with the help of the motion capture system. The tool used to make the reference measurements can be seen in Fig. 4.1(a). Optical markers are attached to the handle so it can be tracked as a rigid body. Using the Motive software, the pivot point of a rigid body can easily be assigned to the position of a marker. This allows setting the pivot point to the tip of one of the attached screws. After removing the marker, the tip can then be used as a calibrated probe to sample points in the reference frame of the motion capture system. The calibrated probe is then used to mark two points on an axis to compute the axis's position and orientation. This procedure is repeated several times, and the average of the measurements is used as a reference. To increase the accuracy of the measured direction vector, additional measurements are collected of moving the probe along the surface of the axis and then fitting a line through it. As one can never mark the exact center of the axis, there are still ambiguities in these measurements. However, this still gives a good guideline for evaluating whether the joint fits are reasonable. In the next step fitting of the different joint types is tested, and the computed axes are compared to the reference measurement.



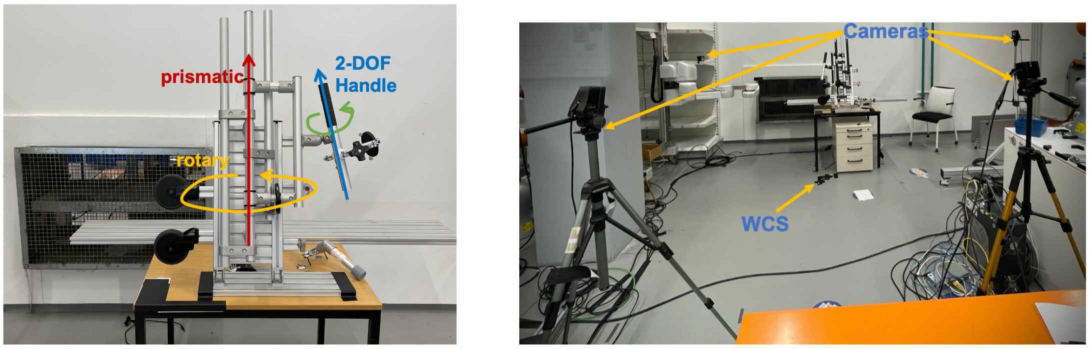
(a) Handle with attached Markers for reference measurements

(b) Sample measurement

Figure 4.1: Calibrated probe

For the different joint/handle types, different metrics are important. The computed axis should be aligned in orientation and position with the reference measurement for a rotary joint. For prismatic joints, an offset of the axis is non-important. Therefore, only the orientational discrepancy is evaluated. For handles, like for rotary joints, both position and orientation are important.

The first experiments on the accuracy of the joint fits are run on the test bench Fig. 4.2(a) we set up that includes different joint types and is re-configurable to change the joint parameters. For evaluation, the central mechanism is used. It includes a cylindrical joint (prismatic and rotary joint which share an axis) with an attached 2-DoF handle. The inclination of the test bench can be adjusted to change the direction vector of the axes. Furthermore, the joint limits of the cylindrical joint can be adjusted, and the orientation of the handle can be reconfigured. An advantage of using this mechanism is that it allows easy measurement of the joint axes, as they are straight bars that are fully exposed. Next to this mechanism, a standard cabinet with multiple prismatic drawers, seen in Fig. 4.5(c) , is used for assessing accuracy. After assessing the accuracy for single joint fits, the full measurement procedure is tested on the two previously mentioned mechanisms and the standard door depicted in Fig. 4.5(b). This is only a qualitative assessment to test whether the implemented procedure works and the models are built correctly.



(a) Annotated test stand

(b) Motion capture setup

Figure 4.2: Evaluation setup

4.2 Experiments

4.2.1 Identifying the Gripper Tip

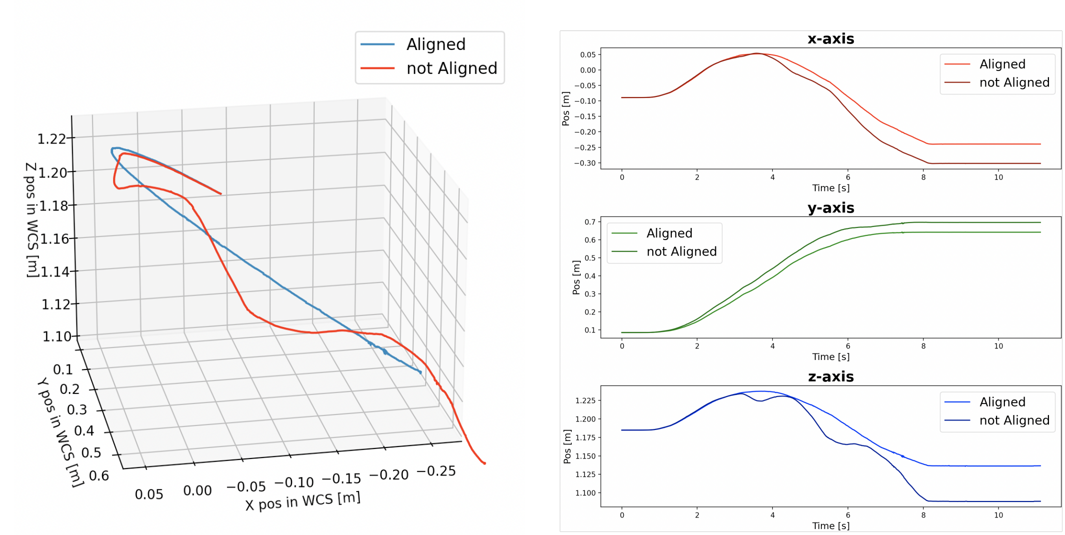
The first set of tests should assess the effect of a misaligned pivot point and if accuracy can be increased with the calibration procedure described in Sec. 3.2.4. When the default pivot-point location, as placed by Motive, is used, the pivot-point is placed in the geometric center of the tracking markers. We can already see that strong deviations can occur if the gripper orientation changes during the measurement. An example trajectory is given in Fig. 4.3. To evaluate how much the calibration improves the fitting accuracy in situations where the gripper orientation is changed during the measurement, a series of measurements is conducted where both the calibrated and the non-calibrated data is recorded. Then joint fits for both are computed and compared to the ground truth. The measurements are conducted on the rotary joint of the testing mechanism; as the ground truth can be accurately estimated on this joint, it has a wide articulation range and is relatively big, which allows focusing only on the investigated effect.

The results of these measurements are shown in Tab. 4.1. What can be seen is that the orientational error with the calibrated tool tip is significantly lower than the orientational error without calibrating the measurement tool. A similar effect can

Meas. series	orient. error	orient. error std. dev.	pos. error	pos. error std. dev.
Calibrated tool tip	0.539°	0.184°	0.0082 m	0.003081 m
Not calibrated	2.260°	1.178°	0.0114 m	0.004669 m

Table 4.1: Results for fitting of rotary joint with and without aligned pivot-point

be seen for the positional error; however, the difference is much smaller here, as the positional accuracy is already fairly good for measurements without an aligned pivot point. One can also see that the standard deviation for both orientational and positional error is significantly lower for the calibrated gripper, which shows that the joint fitting is a lot more stable. All in all, this is evidence that the effect of reorienting the gripper during measurements is reduced by calibrating the gripper using the presented calibration scheme.



(a) Deviations in 3D trajectory with and without aligned pivot-point

(b) Deviations in 2D trajectory with and without aligned pivot-point

Figure 4.3: Influence of non-aligned pivot-point on measurements

4.2.2 Performance with Outliers

In the next step, the effectiveness of using RANSAC to detect gross outliers is tested. This is done by introducing deviations from the joint trajectory during articulation. Again, the evaluation stand is used. It allows accurately computing the ground truth; furthermore, the cylindrical joint makes it easy to add deviations to the joint trajectory during articulation, as one can articulate the respective other DoF. Tests are conducted on both the prismatic and the rotary components of the joint. A relatively aggressive parameterization for the RANSAC is chosen. As the motion capture measurements are normally very clean, the allowable deviations from the model are in the range of a few millimeters. For the measurements, the respective joint is articulated, while operating the joint, deviations are introduced by articulating the second DoF of the joint. Afterward, joint fits are computed with and without RANSAC and compared against the ground truth.

The results of the experiments can be seen in Tab. 4.2. It can be seen that for the rotary joint, the orientational error is significantly reduced by filtering the measurements using RANSAC. The standard deviation of the orientational error is also significantly reduced, which indicates that the fitting is a lot more stable. Interesting is that the positional errors for the fit without RANSAC are better than for the fit with RANSAC. This is probably linked to RANSAC sometimes filtering out a significant number of measurements. As the deviations for the rotary joint are along the axis, the projections into the plane done during circle fitting still lead to those points being very close to the true circular arc, even though the plane might be inaccurate. The model accuracy for the presented scenarios should be sufficient for a robot to interact with these objects, given that there is a control scheme in place that can regulate out small inaccuracies.

In general, we can say that using RANSAC, significant (obvious) outliers can be filtered from the measurements, which allows the fit to be more stable against measurement errors that could easily occur when a human operator articulates a mechanism. However, the robustness of this method should be further improved. This could be done in multiple ways, either by improving the parameterization of the RANSAC, maybe introducing different parameterizations for different size mechanisms. The operator could also be taken further into consideration by showing a warning when a certain percentage of the data is filtered out so that edge cases where the RANSAC filters too many measurements can be accounted for.

4.2.3 Performance on Different Mechanisms

Next, the accuracy of the single joint fits is evaluated with a calibrated gripper, and RANSAC filtering. On the test mechanism identification is tested on the rotary

Mechanism		orient. error	orient. error std. dev.	pos. error	pos. error std. dev.
Rotary	RANSAC	1.039832°	0.880033°	0.006717 m	0.001858 m
	No RANSAC	6.065697°	3.301993°	0.004962 m	0.001617 m
Prismatic	RANSAC	1.082944°	0.540166°	-	-
	No RANSAC	12.659371°	1.136343°	-	-

Table 4.2: Comparison of fitting with and without RANSAC for stark outliers

and prismatic joint, the attached 2-DoF handle and the central axis evaluated as a 2-DoF handle. On the cabinet, tests are conducted on the prismatic joint and 2-DoF handle of the uppermost drawer.

The results of these tests are presented in Tab. 4.3. From the data, one can see that the fitting gives accurate results for all tested joint types. Noticeable is that the fitting of the prismatic component on the test mechanism is worse than that of the rotary component. This can presumably be explained by the mechanics of the mechanism, which makes it harder to move the prismatic component without disturbances, as the mechanism has a cylindrical joint at its core, rather than a singular prismatic and a singular rotary joint. Furthermore, the prismatic joint does not move perfectly smoothly, resulting in a shaky motion during articulation. The mentioned problems result in minimal and short-scale deflections. These are either filtered out by RANSAC, leading to large chunks of the data being filtered out, or unrecognized. In both scenarios this leads to worse fitting accuracy, as either only a small subset of data is considered, or there are non-Gaussian outliers in the data that negatively affect the least-squares fitting. For the handle fitting, the orientational accuracy is about as good as that of the prismatic joint. The positional accuracy is problematic when compared to the diameter of the handle, which is 3 cm. To test whether this can be explained by the mechanics of the mechanism, the interaction or maybe even the ground truth, which could be less accurate due to the handle being much shorter than the central bar, another test is conducted where instead of using the dedicated handle, the central axis is used. What can be seen is that this fit is more accurate than the fit of the other handle, in both positional and orientational error. This shows, that the inaccuracy of the handle fitting might not only be caused by the model fitting, but might be influenced by one of the other mentioned factors. This hypothesis is further supported by the orientational accuracy of the cabinet handle identification. Still, even for this ”replacement” handle, the positional accuracy of the handle fitting is lower than for the rotary joint.

Mechanism	orient. error	orient. error std. dev.	pos. error	pos. error std. dev.
Rotary joint	0.379264°	0.164402°	0.005217 m	0.002544 m
Prismatic joint	1.264748°	0.530295°	-	-
Handle	1.030192°	0.534674°	0.014495 m	0.010748 m
2nd Handle	0.646315°	0.550131°	0.007056 m	0.001096 m
Cabinet drawer	0.048627°	0.014971°	-	-
Cabinet handle	0.564795°	0.010581°	0.015051 m	0.001716 m

Table 4.3: Results for fitting different joints with aligned pivot-point

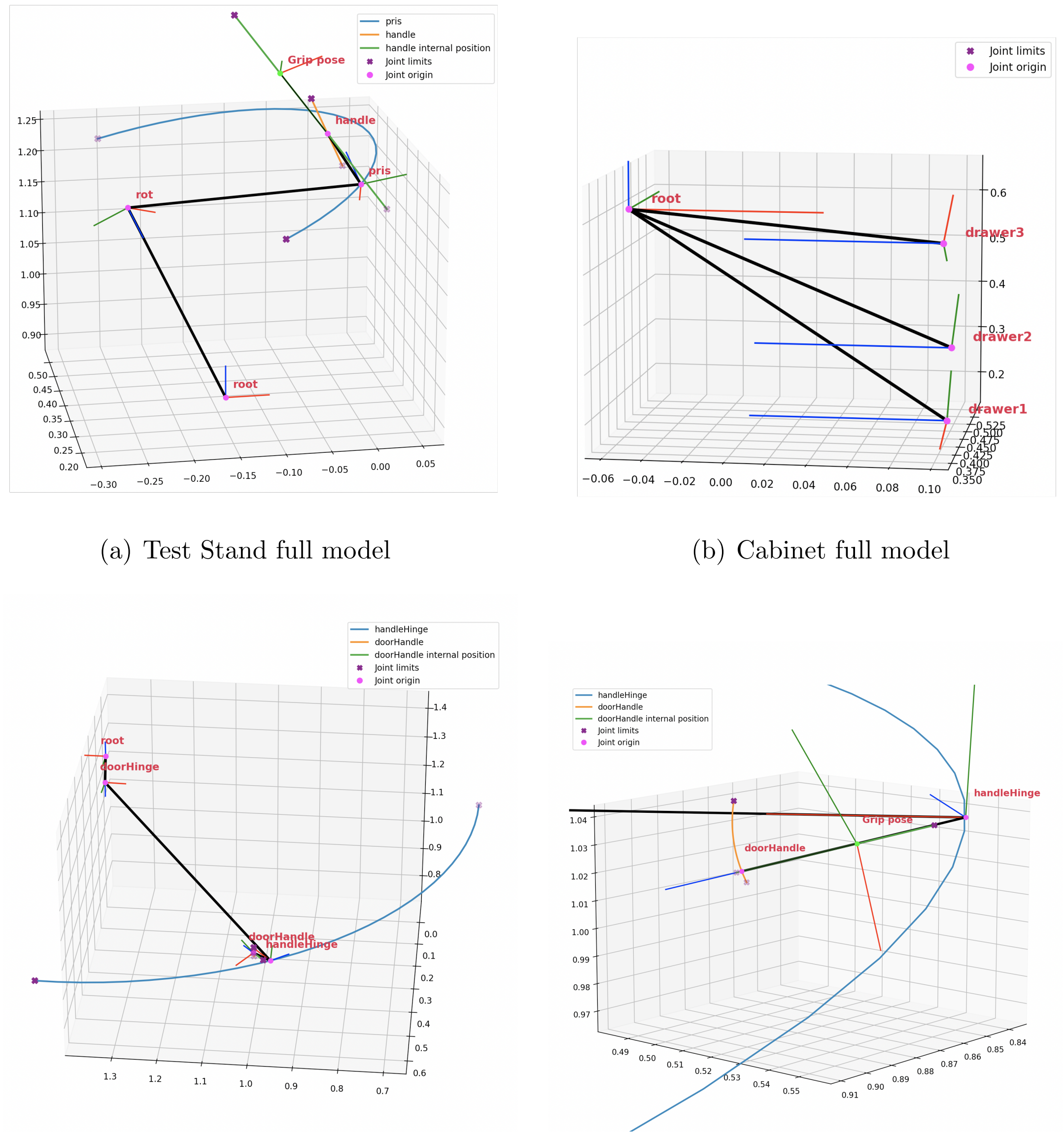
We can see that the fitting of both the prismatic axis and the handle is accurate in terms of orientational error for the cabinet measurements. Especially the fitting of the prismatic axis is very accurate. This is probably due to the joint trajectory being relatively long. The orientational accuracy of the handle fit shows that the procedure also accurately works for relatively small handles. However, considering that the diameter of the handle is about 1.5 cm, the positional error of about 1.5 cm is problematic. This is mainly caused by the short articulation range observed of the handle's circular component. The problem is that the handles are very close to each other and the cabinet itself. Therefore, during the rotary movement, the gripper's jaws quickly come into contact with either another handle or the cabinet, which leads to a short observed articulation range. With such small articulation ranges, the identification of the plane becomes inaccurate, as considering some noise and inaccuracies, there are many planes that present a good fit to the data.

All in all, it can be concluded that the fitting procedure mostly provides accurate results. The major problem is scenarios where, either due to the mechanics of the gripper or due to the mechanism itself, the articulation range is short, or it is not possible to follow the joint trajectory accurately.

4.2.4 Identification of Complete Models

In the next step, an identification of the complete kinematic chain of the different mechanisms is conducted. Figure 4.4(a) shows a visualization of the computed model for the test mechanism. As each of the joints is fit independently of the other joints, the errors are again in the same range as those computed on the initial testing. In

Fig. 4.4(a), the general kinematic structure is represented correctly as a kinematic chain of the model root, rotary, and prismatic joint with a 2-DOF handle in the end. The joint frames are aligned according to the respective articulation directions and joint limits. The results of the model fitting on the cabinet can be seen in Fig. 4.4(b). The fitting of the main door hinge is easily done; however, identifying the smaller rotary mechanism of the door handle that only offers a small articulation range (c.a. 20°) is unstable. In some tests, the result is satisfying; in some tests, the result is far off. The found kinematic structure can be seen in Fig. 4.4(c), and a zoomed-in view can be seen in Fig. 4.4(d).



(a) Test Stand full model

(b) Cabinet full model

(c) Door full model

(d) Door handle zoomed in

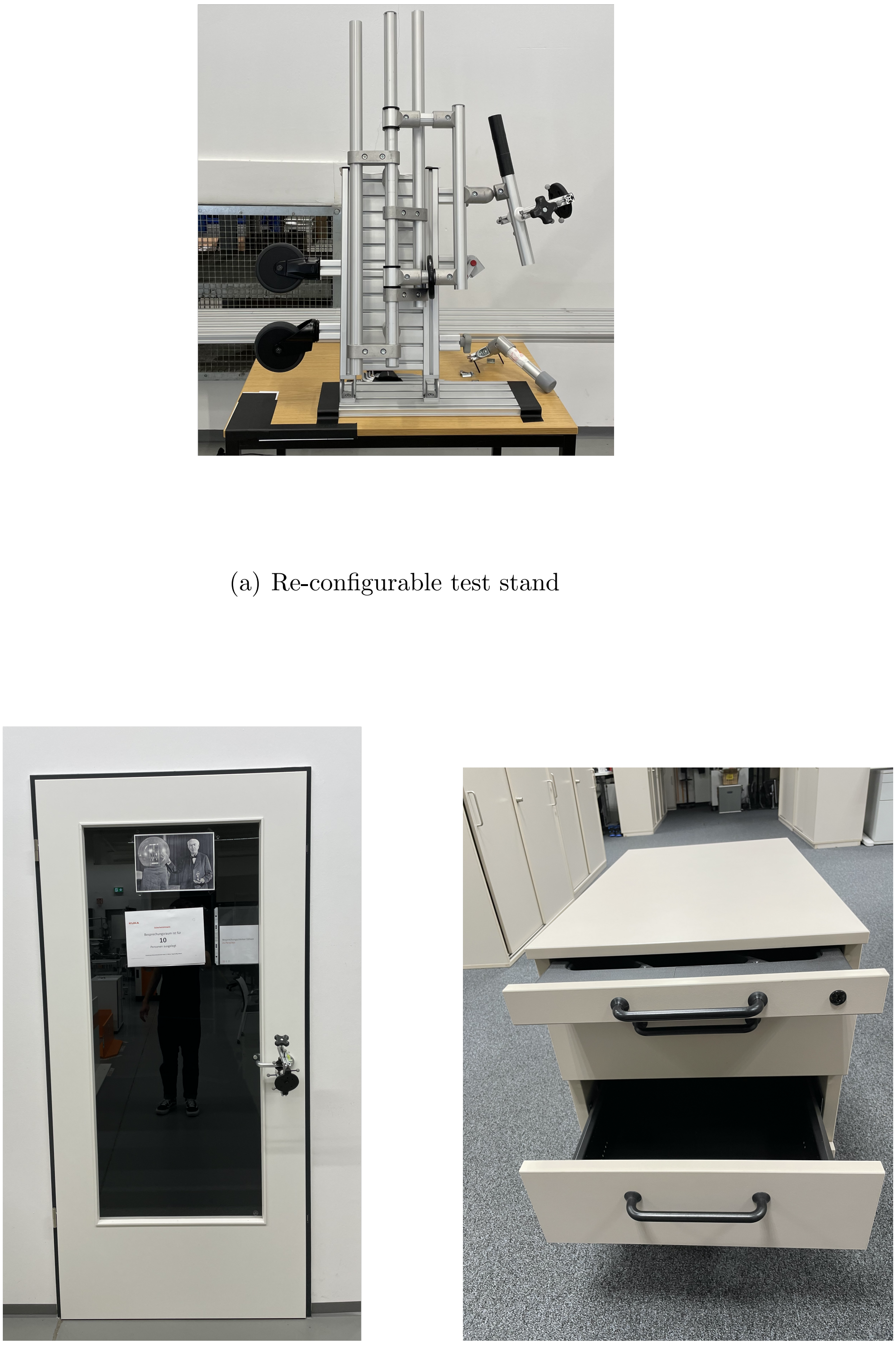
Figure 4.4: Plots of identified models for (left to right, top to bottom) test mechanism, cabinet, door

4.3 Discussion

The presented experiments show that identifying kinematic models of articulated objects is possible with the presented framework. This works for simple kinematic chains (test mechanism, door) and kinematic trees (cabinet). The implemented procedure allows a human operator to identify the model of an articulated object piece by piece in a guided procedure. However, there are still problems, mainly with the robustness of the implementation. As described for the interaction with the cabinet, there are limits to what types of mechanisms can be interacted with using the current gripper design. Due to the width of the gripper jaws and the "fingers" being completely recessed between the two jaws, it is hard to grasp and collect data for any mechanism where the handles are rather close to each other. Identifying expansive joints like the hinge of a door or the prismatic joint of a drawer works reliably, but there are problems with smaller joints, like those found on door handles.

In general, for situations where it is impossible to get a good, stable trajectory, it would be useful to implement a fall-back solution that the operator could use to increase the accuracy of the measurement. This is especially important for reliable identification of handle models, as these are normally small-scale, high fidelity interactions that are hard to get right with the current gripper design and fitting procedure. Another way to add robustness would be to use knowledge of the gripper and articulated object kinematics to estimate errors in the measurements. For example, knowledge of the handle model could be used to estimate when slipping along the handle occurs. Like Martin-Martin et al. [MMB17] propose, adding in additional sensors, like an FT sensor, not only allows identifying additional attributes but can also improve the identification of kinematic models, as errors invisible in one sensor domain are often apparent in another one. Right now, full identification of the test mechanism already takes about ten minutes. This could be decreased through a better user interface that allows users to interact with the measurement application through input on the gripper rather than a laptop. Still, not every task initially not easily solvable should be given to the operator to specify based on knowledge. For example, after specifying the connectivity of the articulated object, the types of joints involved could, most probably, also be identified by the software. In most cases, a complicated probabilistic framework like the one implemented by Sturm [Stu11] is not even needed. Computing fits for every model and then comparing the root-mean-squared errors of all the fits would probably be sufficient in most cases.

As illustrated in Sec. 2.3, some methods can use generalized models of articulated objects for robot manipulation. These could be adapted to work with our model formulation, as they normally only need to query the model for a joint trajectory or a Jacobian. A limitation is that these methods are still mostly limited to single-joint mechanisms. In the current model identification implementation, only the model root, joint frames, and handle frames are modeled and tracked. The operator could use any point on the articulated object to actuate a particular joint during the model identification process. Therefore, there is still ambiguity in where a robot should actually grasp an articulated to actuate a specific joint. For example, when opening a door, we grab the handle, both for unlatching the door by pressing the handle and articulating the door hinge. For this reason, the current modeling approach models handles, as these can serve as a defined gripping point on the mechanism for manipulation. This formulation is not perfect, as it discards many of the ways we actually interact with a mechanism; for example, we sometimes push on the door itself to close it rather than on the handle. As the measurements during model identification resemble robot end effector poses, it is also conceivable to use these as reference trajectories for a robot to articulate the single joints. A more sophisticated approach would be to add marking other positions on the articulated object to associate them with a specific joint during model fitting to provide multiple grasping positions. However, it has to be evaluated whether this is sufficient to deal with the issue.



(a) Re-configurable test stand

(b) Door

(c) Cabinet with prismatic drawers

Figure 4.5: Mechanisms on which tests are conducted

Chapter 5

Conclusion

5.1 Achievements and Limitations

This thesis, proposes and implements a proof of concept system for human-assisted identification of articulated objects. It can be seen that it is possible to find key attributes of an articulated object with the proposed framework. However, the current implementation is limited to identifying the simplified kinematic structure of articulated objects. Furthermore, the current implementation lacks robustness. Depending on the quality of the grip, the size, and the shape of the mechanism, the system performance varies from very accurate to passable. This is both a mechanical as well as a software problem. A lot more attributes could be identified, which would make the system more useful as a universal toolbox for identifying various articulated objects. The current implementation lays the groundwork for implementing these extensions. This work can be seen as a base implementation from which one can start more detailed investigations into the different components to improve the overall performance(gripper, model fitting, data acquisition, user interaction).

5.2 Outlook

By expanding the model formulation to allow modeling of joint dependencies, hybrid joints, and joint inaccuracies, and with added robustness and versatility, this concept can be developed into a powerful platform for quickly and intuitively modeling articulated objects. As illustrated in Sec. 4.3, there are multiple avenues that can be pursued to improve robustness and versatility. Through an improved user interface the ease-of-use of the system can be greatly improved, which decreases the time it takes to identify kinematic models, and also allows gathering additional user insights. Another interesting expansion to the system is to explore the field of transfer learning. One example would be, to directly transfer the articulation measurements to a robot, and have the robot learn a model by itself by following the measured trajectory. There is mounting evidence that humans interact with their environment through learned models that allow predicting expected future outcomes [LPAH19]. Therefore, we are convinced that by learning and using expressive models of the environment, a lot of progress can be made towards more autonomous robots.

List of Figures

Bibliography

[BHB13] Felix Burget, Armin Hornung, and Maren Bennewitz. Whole-body motion planning for manipulation of articulated objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 1656–1662, 2013. doi:10.1109/ICRA.2013.6630792.

[BK19] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge University Press, 2019. doi:10.1017/9781108380690.

[CCL10] Sachin Chitta, Benjamin Cohen, and Maxim Likhachev. Planning for autonomous door opening with a mobile manipulator. In *2010 IEEE International Conference on Robotics and Automation*, pages 1799–1806, 2010. doi:10.1109/ROBOT.2010.5509475.

[Com] The Qt Company. Qt creator. https://www.qt.io/product/development-tools. Accessed: 2021-08-20.

[ETB13] Felix Endres, Jeff Trinkle, and Wolfram Burgard. Learning the dynamics of doors for robotic manipulation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3543–3549, 2013. doi:10.1109/IROS.2013.6696861.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 06 1981. doi:10.1145/358669.358692.

[GPL17] Cuiping Guo, Junhuan Peng, and Chuantao Li. Total least squares algorithms for fitting 3d straight lines. *International Journal of Applied Mathematics and Machine Learning*, 6:35–44, 05 2017. doi:10.18642/ijamml_7100121818.

[HAJ06] T. Hardeman, R. G. K. M. Aarts, and J. B. Jonker. A finite element formulation for dynamic parameter identification of robot manipulators. 16:21–35, 08 2006. URL: https://doi.org/10.1007/s11044-006-9010-x.

[HNOS15] Karol Hausman, Scott Niekum, Sarah Osentoski, and Gaurav S. Sukhatme. Active articulation model estimation through interactive perception. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3305–3312, 2015. doi:10.1109/ICRA.2015.7139655.

[Hun07] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.

[JK09] Advait Jain and Charles C. Kemp. Pulling open novel doors and drawers with equilibrium point control. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 498–505, 2009. doi:10.1109/ICHR.2009.5379532.

[JN19] Ajinkya Jain and Scott Niekum. Learning hybrid object kinematics for efficient hierarchical planning under uncertainty. *CoRR*, abs/1907.09014, 2019. URL: http://arxiv.org/abs/1907.09014, arXiv:1907.09014.

[JNR$^+$10] Advait Jain, Hai Nguyen, Mrinal Rath, Jason Okerman, and Charles C. Kemp. The complex structure of simple devices: A survey of trajectories and forces that open doors and drawers. In *2010 3rd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 184–190, 2010. doi:10.1109/BIOROB.2010.5626754.

[KOT15] Johannes Kulick, Stefan Otte, and Marc Toussaint. Active exploration of joint dependency structures. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2598–2604, 2015. doi:10.1109/ICRA.2015.7139549.

[LPAH19] Marcus Lewis, Scott Purdy, Subutai Ahmad, and Jeff Hawkins. Locations in the neocortex: A theory of sensorimotor object recognition using cortical grid cells. *Frontiers in Neural Circuits*, 13:22, 2019. URL: https://www.frontiersin.org/article/10.3389/fncir.2019.00022, doi:10.3389/fncir.2019.00022.

[LST12] John Lygeros, Shankar Sastry, and Claire Tomlin. *Hybrid Systems: Foundations, advanced topics and applications.* under copyright to be published by Springer Verlag, 2012.

[MMB17] Roberto Martin-Martin and Oliver Brock. Building kinematic and dynamic models of articulated objects with multi-modal interactive perception. 03 2017.

[Nat] NaturalPoint. Optitrack primex13. https://optitrack.com/cameras/primex-13/. Accessed: 2021-08-10.

[NOAB15] Scott Niekum, Sarah Osentoski, Christopher G. Atkeson, and Andrew G. Barto. Online bayesian changepoint detection for articulated motion models. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1468–1475, 2015. doi:10.1109/ICRA.2015.7139383.

[PCA16] *Fitting a Circle To Cluster of 3D Points*, 08 2016. URL: https://meshlogic.github.io/posts/jupyter/curve-fitting/fitting-a-circle-to-cluster-of-3d-points/.

[PWT15] Sudeep Pillai, Matthew R. Walter, and Seth Teller. Learning articulated motions from visual demonstration, 2015. arXiv:1502.01659.

[RBB20] Adrian Röfer, Georg Bartels, and Michael Beetz. Kineverse: A symbolic articulation model framework for model-generic software for mobile manipulation. *CoRR*, abs/2012.05362, 2020. URL: https://arxiv.org/abs/2012.05362, arXiv:2012.05362.

[RSF13] Eric Rohmer, Surya P. N. Singh, and Marc Freese. Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. www.coppeliarobotics.com.

[RSP$^+$12] Thomas Rühr, Jürgen Sturm, Dejan Pangercic, Michael Beetz, and Daniel Cremers. A generalized framework for opening doors and drawers in kitchen environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 3852–3858, 2012. doi:10.1109/ICRA.2012.6224929.

[RW05] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2005.

[Sch78] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461 – 464, 1978. doi:10.1214/aos/1176344136.

[SJS$^+$10] Jürgen Sturm, Advait Jain, Cyrill Stachniss, Charles C. Kemp, and Wolfram Burgard. Operating articulated objects based on experience. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2739–2744, 2010. doi:10.1109/IROS.2010.5653813.

[SSP$^+$09] Jürgen Sturm, Cyrill Stachniss, Vijay Pradeep, Christian Plagemann, Kurt Konolige, and Wolfram Burgard. Towards understanding articulated objects. In *Proc. of the Workshop on Robot Manipulation at Robotics: Science and Systems Conference (RSS)*, 06 2009.

[Stu11] Jürgen Sturm. *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots*. PhD thesis, University of Freiburg, Germany, 05 2011.

[SV08] Sven Serneels and Tim Verdonck. Principal component analysis for data containing outliers and missing elements. *Computational Statistics & Data Analysis*, 52:1712–1727, 02 2008. `doi:10.1016/j.csda.2007.05.024`.

[YSL21] Yongqiang Yu, Ran Shi, and Yunjiang Lou. Bias estimation and gravity compensation for wrist-mounted force/torque sensor. *IEEE Sensors Journal*, pages 1–1, 2021. `doi:10.1109/JSEN.2021.3056943`.

License