Online Bayesian Segmentation during Kinaesthetic Teaching with Dynamic Models

handed in
MASTER'S THESIS

Bachelor of Sciene Uhl Fabian

born on the 22.12.1989
living in:
Weltistraße 98
81477 Munich
Tel.: 08985670296

Human-centered Assistive Robotics
Technical University of Munich

Univ.-Prof. Dr.-Ing. Dongheui Lee

Supervisor:	Dr.-Ing. José Ramon Medina Hernandez
Start:	16.12.2019
Intermediate Report:	23.08.2020
Delivery:	29.09.2020

TECHNISCHE UNIVERSITÄT MÜNCHEN

Human-centered Assistive Robotics

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

September 21, 2020

MASTER'S THESIS
for
Fabian Uhl
Student ID 03651245, Degree EI

Online Bayesian Segmentation during Kinaesthetic Teaching with Dynamic Models

Problem description:

Kinaesthetic robot teaching is the most popular learning by demonstration interface. Although many approaches try to mimic demonstrated motions as accurately as possible, it is well known that human demonstrations during teaching are far from optimal. In fact, non expert teachers deviate significantly from any expected optimal behavior. Instead of targeting optimal imitation, this thesis aims for estimating human intention on a higher abstraction level based on continuous robot motion and force signals. We assume that the human-robot physically coupled dyad behaves as a hybrid switching controller comprising a set of high level operations and switching criteria. To identify those during teaching, this thesis will explore the combination of online Bayesian change point detection algorithms with task-specific linear dynamic models and switching conditions. The proposed method should automatically extract a sequence of simplified and parameterized operations and potential termination conditions. The approach will be evaluated on a Franka Emika Panda robot equipped with a force sensor in a user study in order to evaluate is intuitiveness and efficacy w.r.t. state of the art kinaesthetic teaching methods.

Tasks:

- Summarize related work of segmentation and classification of robot tasks.
- Evaluation of Bayesian change point detection and segmentation algorithms with different robot task / dynamic models. [1] [2]
- Validation of the classification by different robotic task models.
- Implementation of a online segmentation algorithm that directly segments robot data into higher level operations and their parameterization.
- Verification of the proposed approach in a user study comparing it with the commercially available Franka Emika teaching method *Desk*.

Bibliography:

[1] Ryan Prescott Adams. Bayesian online changepoint detection. Technical report, 2007.

[2] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters. Learning movement primitive libraries through probabilistic segmentation. *International Journal of Robotics Research*, 36(8):879–894, July 2017.

Supervisor:	Dr.-Ing. José Ramon Medina Hernandez
Start:	6.12.2019
Intermediate Report:	23.08.2020
Delivery:	29.09.2020

(Dongeui Lee)
Prof.

I confirm that this is my own work and I have documented all sources and material used.

München, 29.09.2020
(Place, Date)

(B.Sc. Fabian Uhl)

Abstract

Kinaesthetic robot teaching, where a robot is physically guided by a human demonstrator, is a fast robot programming paradigm especially suitable for the flexible automation problems that arise in today's industrial manufacturing landscape. Understanding a robot application as a sequence of subtasks, the *segmentation* of captured demonstrations and the *classification* of identified segments are the first key steps towards an fully automated parameterization of a robot program. These two steps are typically decoupled from each other and even consider different optimization criteria. This thesis aims for a segmentation approach that merges the classification and segmentation steps. Specifically, in this work we present a segmentation technique that incorporates knowledge about the interaction dynamics and simple human models into its segment model. This enables a joint segmentation and classification approach based solely on estimated control parameters. Experiments on a 7 degrees-of-freedom manipulator equipped with a force sensor on its wrist show promising results in a pick and place scenario and a scenario where a soap dispenser is pressed. Finally, we also validated the applicability of the segmentation algorithm in an online fashion during the teaching process obtaining comparable results.

Contents

Chapter 1

Introduction

Among all robot programming paradigms, *kinaesthetic teaching* has become a popular method to quickly prototype solutions with robot manipulators, especially in the field of industrial manufacturing. It aims to take advantage from human cognitive capabilities, namely the ability to understand a task, its structure, goals and conditions and transfer them to the robot through demonstrations. Human workers are extremely efficient and flexible: Thanks to their deeper task understanding, they are able to optimize a task for any desired criteria on the fly and also to detect errors during the task execution. To achieve the same level performance and flexibility on a robot from demonstrations, solely mimicking human motions is not sufficient; the recognition of the higher level human intention from demonstrated behavior is key. This becomes a very challenging task when one considers all possible semantic interpretation of a motion captured by a limited set of robot signals. The task segmentation problem is an Divide and Conquer approach, where the task is partitioned in to sub-tasks with different semantic interpretation. The main challenge of this problem is to represent and detect these subtasks at the same level of abstraction that the human teacher might have in mind.

Current approaches separate the segmentation and the segment classification into two sequential steps. First, a *segmentation algorithm* identifies a set of *changepoints*, i.e. specific points in time that define segments of the time series that represents the observed behavior. With these segments a *segment classification algorithm* performs then a semantic evaluation in order to extract a definition of the identified substask. However, these two steps are typically decoupled from each other and even have different optimization goals or heuristics. For instance, the segmentation algorithm might exploit no information about the possible set of actions that the classification algorithm considers.

To bypass such a loss of information, this thesis aims for a segmentation approach that combines the classification and the segmentation task into one step. Specifically, in this work we present an online segmentation technique that incorporates knowledge about the interaction dynamics and simple human models into its segment model. This enables a joint segmentation and classification approach based solely on estimated control parameters. Experiments on a 7 degrees-of-freedom manipulator equipped with a force sensor on its wrist show promising results.

1.1 Related Work

The segmentation of time series data is a omnipresent problem in a wide variety of fields. In the following sections, we will present divided into event-based, and model-based current segmentation approaches.

1.1.1 Event-based Segmentation

Events are observable patterns derived from heuristics and physical boundaries are events . The definition of the event also determines the segment definition, which means it specifies the requirements such as the input space dimensions. Especially, the generalizable and scalability of the segmentation is derived from the quality of the used event. For instance, signal manipulation such as noisy sensor signals can emit fake segments, which damages the stability of the segmentation. In most cases the definition of the event is fixed, therefore further training becomes not necessary and by adding the low computational effort of the observation and pattern matching, the segmentation algorithm can be applied in an online application case. Without further information, the verification evaluates the quality of the defined event. Due to the fact, that most segmentations approaches only use the event segmentation as the pre-segmentation step and the verification is applied on the final results, the verification contains a combination of both techniques.

In a robotic scenario and in the case of movement primitive segmentation, the zero velocity crossing event is widely used. In many papers, this fast technique is used in a preprocessing step in form form of a pre-segmentation. The result is a reduced set of possible change points with a more manageable size. Therefore, the main segmentation part eliminate possible instabilities of these technique, which can be for instance fake event generated due to signal manipulation. [FMJ02] [LKK16] [LNMP17] [MTS12] Another approach uses the local minima or maxima in the joint angle space of a humanoid robot as events to detect movement primitives of observed gestures [CB04]. Supported with a wearable inertial measurement unit (IMU) sensor, an event can be defined in the force space to segment the individual strides a walking human [ZKJ+13].

1.1.2 Model-based segmentation

Approaches from this category use a set of underlying models to identify piece-wise common features in the input space. By assigning of the sections in the observation to these underlying models, the segmentation is obtained. The quality of the segmentation is based on the accuracy of the underlying model and the input space. To simplify the evaluation of the segmentation, the underlying model formulation is often based on a probabilistic model, such as the already introduced Partition Product Model. In the case of model-based segmentation techniques, the segment definition is completed by the definition of the underlying models. The form of such a model definition can be a set of model parameters or a library of patterns. The generalizability and scalability of segmentation highly depend on the definition of the individual models. For instance, the size of the model library and the correlation among the models are further significant values for the generalizability and scalability of segment definition. The main task of the segmentation algorithm is to find the best assignment of the section of the input space to the models. In the case of a probabilistic model formulation, the segmentation is achieved by maximizing the likelihood. Depending on the complexity and size of the model library and input space, the computational effort allows an offline, semi-online, or online application case. In the case of an adaptive model, the defined model update function has to be taken into account, especially in the context of the computational effort. In general, more complex models allow more accurate and detailed segmentation, where simple models are more suited for the identification of underlying structures in the input space. The segmentation definition and algorithm can be verified by comparing the results of a manually labeled data set.

A simple model for segmentation of movement primitives can be defined in the mean squared or mean of the distance of the position trajectory. As model parameter suits a mean position and by assigning all position points to a limited amount of models, the segmentation generates a reduced sequence of positions [Pom00] [JGPV17].

Further, the segmentation can be reformulated as into a classification problem. For this, every model is represented by a class, and the defined decision rule is used to assign the input space to the models [YZL$^+$19].

Another approach uses a library of generic movement primitives, which can be generated and trained supported by for instance Probabilistic Dynamic Movement Primitives (ProDMP) and the segmentation will be achieved by recognizing the movement primitives [LNMP17]. By a probabilistic formulation, the recognition becomes an optimization, which allows the usage of Expectation maximization (EM), or maximum a posterior (MAP) with a problem formulation. By pre-segmentation, a reduced set of possible change points is generated, and therefore, the optimization space is reduced dramatically. This reduction generates the trade-off, that the best solution isn't part of the optimization space.

Bayesian Change point detection

The Bayesian change point detection computes the probability of a change point for every timestamp based on the Bayesian inference method and a defined probabilistic model. The segment definition contains the definition of the model, which is extended by the requirement of an adaptive Bayesian prior formulation. The original formulation contains a normal distributed with non-specified conjugate prior to the model parameter [Fea06], but the Bayesian change point detection algorithm is able to work with every distribution from the exponential family or other distribution, for which a conjugate prior formulation exists. Further, the model can also be extended with for instance a linear model by including the linear regression formulation [XM07]. Based on the Partition Product model is the Bayesian change point detection able to perform in an online application case without any pre-segmentation. This is possible due to the incremental model update function, which updates the segmentation with every new incoming data point. In an offline application, the number of possible segmentations increases squared with the size of the data set, which leads to a high computational effort. To improve the computational efficiency, Paul Fearnhead introduced an algorithm, which uses a recursive dynamic programming approach and a pruning threshold to reduce the number of calculation to the most probable hypotheses [Fea06]. Due to the formulation of the segmentation result as probability of a change point at ever time stamp, the verification can be applied in different sensibilities for a change point. Additionally, the define model and the Bayesian change point detection approach provide different hyper parameter, which allow a fine tuning of the result.

The Bayesian's change point detection demonstrates good performance in other research areas, for instance, the research of the returns of Dow Jones Industrial Average to detect significant events [Fea06]. Annual global surface temperature anomalies are successfully detected in the data from the National Oceanic and atmospheric administration (NOAA) and National Centres Data Center (NCDC) by the Bayesian change point detection extending the model with a linear regression formulation [Rug13]. An online variation of the Bayesian change point detection algorithm is used, to identify segments of activity corresponding to the different states of a epileptic brain. For that, the electrocardiography (ECoG) of an epileptic patient is used [MKA13].

1.2 Outline

The remainder of this thesis is structured as follows. In Chapter 2, the used techniques and their implications are described. The proposed approach is introduced in Chapter 3. Chapter 4 presents all considered scenarios and preliminary results. Finally, in the Chapter 5 the achieved results is summarized.

Chapter 2

Background

This chapter exposes the necessary background on Bayesian Changepoint detection (BCPD) algorithm, robot modeling, and control that this thesis builds upon.

2.1 Bayesian changepoint detection

In this section, we will introduce the BCPD algorithm. The BCPD algorithm uses Bayesian inference and a probabilistic model to compute the probability of every data point to be a proper changepoint. Therefore, the segmentation can be designed by the used probabilistic model and the corresponding model parameter search space.
In the Section 2.1.1, and Section 2.1.2, we will introduce the Product Partition Model (PPM) and the Segment Optimization formulation, which describe the mathematical foundation of the BCPD. In Section 2.1, we will introduce the offline BCPD algorithm and in section 2.1.5 the online Bayesian Changepoint detection (oBCPD) algorthim. Finally, techniques to reduce to computational effort are introduced in Sec. 2.1.6.

2.1.1 Product Partition Model

Initially, we will introduce the definition of a partition, by taking advantage of the structure of the input signal and the definition of the segmentation formulation. The extension of the segment definition by a probabilistic model formulation will allow us the definition of the PPM, which enables the evaluation of the assigning of every possible partition to the defined model parameter. Finally, we will introduce a formulation, which enables us to evaluate a segmentation based on a chosen set of change points and model parameters [QI03] [Dah09]. The PPM is based on the assumption of underlying models, where a model is defined by a feature function. These feature functions, described by the model parameters Θ, generate the observation Y depending on the input signal X. This result in the formulation $Y = f_i(\theta_i, X)$, for the model i and the corresponding feature function f_i. Supported by this definition, the subsequently detecting of feature changes can be interpreted as the switching of the underlying model and thereby, the changing of the model parameter. Additionally, we assume the input signal will has the form of a discrete-time signal, which is commonly given by the sampling of sensor signals and will result in the same form for the observation. Finally, this allows the definition of a partition as a section in the observation $Y_{1:n}$ like $Y_{i:j} : \{y_i, \ \dots, \ y_j\}$, where $1 \leq i < j \leq n$. The segmentation can be represented by the set of changepoints C, which contains the indexes of the first observation of every segment and the index of the final observation like $C : \{1, c_1, \dots, c_k, n\}$ and are sorted chronological. As a segmentation is considered non-overlapping, the set of changepoints C generate the segmentation $S\}$ as the sequence of segments like

$$\begin{aligned} S :& \{s_1, s_2, \dots, s_k\} \\ s_i :& \{Y_{i:i+1} | i \in C \land i \neq n\} \ , \end{aligned} \tag{2.1}$$

where the last segment s_k is extended by the last point y_n
With the product partition distribution formulation [BH92a], the segmentation can be formulated as the product of probabilities like

$$P(Y|S) = P(Y_{s_0})P(Y_{s_1}) \dots P(Y_{s_k}) \ \ = \prod_{s_i \in S} P(Y_{s_i}) \ . \tag{2.2}$$

With the given model probability density $P(y|\theta)$, desired features and characteristics of a underlying model can be described by the model parameter θ. By assuming, that every data point is independent and identically distributed (i.i.d.), the correlation between every data point in a segment and the model parameter θ can be formulated like

$$P(Y_{i:j}, \theta) = \prod_{l=i}^{j-1} P(y_l|\theta)P(\theta) \ , \tag{2.3}$$

where $P(\theta)$ is the model parameter prior. In the context of the segmentation, the model parameter θ become a latent variable and with the marginalisation of the model parameter, the formulation can be written like

$$P(Y_{i:j}|\Theta) = \int \prod_{l=i+1}^{j} P(y_l|\theta)P(\theta)d\Theta \ , \tag{2.4}$$

where Θ is the set of all model parameter [BH92b].
Finally, the probability of given segmentation S can be formulated like

$$\begin{aligned} P(Y_{1:n}|S, \Theta) &= \prod_{s_i}^{S} P(s_i|\Theta) \\ &= \prod_{s_i}^{S} \int \prod_{y_i \in s_i} P(y_i|\theta)P(\theta)d\Theta \ . \end{aligned} \tag{2.5}$$

2.1.2 Segmentation Problem Formulation

The goal of the segmentation is to find the set of changepoints, which separate the input signal into sections, which have the highest matching to predefined criteria. These criteria are defined in model formulation and by assigning the set of model parameters Θ.

Base on the probability of a given segmentation Eq. (2.5) provided by the PPM formulation and the Bayes'theorem, the switched probability of a segmentation given a observation can be written like

$$P(S|Y,\Theta) = \frac{P(Y|S)P(S)}{P(Y)} \ , \tag{2.6}$$

where the segmentation have the form defined in Eq. (2.1) and the segmentation prior $P(S)$ define segmentation features such as a desired average segment length.

Segmentation Search Space

As the search space for the segmentation optimization, every data point is taken into account, and therefore, the power set $\wp$ of every data point generates the set of possible changepoint combinations D. As every segmentation have to cover the complete observation, the first and last data point has to be a part of every possible set of changepoints and thus, the search space is defined like

$$\begin{aligned} D &:= \{y_0, \wp(Y_{2:n-1}), y_n\} \\ D_S &:= \{S(C)|C \in D\} \ , \end{aligned} \tag{2.7}$$

where set of segmentations D_S is generated with the segmentation definition Eq. (2.1).

2.1.3 Segmentation Optimization

With the definition of the search space in Eq. (2.7), the segmentation optimization can be formulated as a maximum likelihood estimation of the segmentation probability Eq. (2.6) and is written like

$$\begin{aligned} S^\star &= \underset{S^\star \in D_S}{\operatorname{argmax}} \ P(S|Y,\Theta) \\ S^\star &= \underset{S^\star \in D_S}{\operatorname{argmax}} \ \frac{P(Y|S,\Theta)P(S)}{P(Y)} \\ S^\star &= \underset{S^\star \in D_S}{\operatorname{argmax}} \ \frac{\prod_{s_i}^{S \in D_S} \int P(s_i|\theta)P(\theta)d\Theta P(S)}{\sum_{S}^{D_S} \prod_{s_i}^{S} \int P(s_i|\theta)P(\theta)d\Theta P(S)} \\ &, \end{aligned} \tag{2.8}$$

where $S^\star$ is the optimal segmentation for the observation Y.

The size of the observation n is depending on the mean resolution r and the duration of the observation t_n. If we combine this linear growing with the squared growing of the power set D, the computational effort of segmentation will be highly increased for longer observations $O((rn)^2)$. So, especially the reduction of the observation size is one of the first approaches, to limit the computational effort and thus, achieve the computational performance for an online segmentation.

2.1.4 Offline Segmentation Algorithm

The offline Bayesian changepoint detection algorithm solves the segmentation problem formulation introduced in Sec. 2.1.2 , by using a recursive formulation of the forward-backward algorithm.
The priors introduced in the next two sections allow you to customize the segmentation according to an expected segment length distribution or the desired number of changepoint numbers. The offline BCPD algorithm consists of three steps. In the first step, every possible segment likelihood in the given observation is computed and stored. In the next step, the likelihood of every possible segmentation is computed based on the forward recursion technique. In the final step, the changepoint probability is computed as exclusively dependent on the considered location by marginalizing the number of changepoints.

Segment Length Prior

The distribution of the resulting segment lengths can be controlled with the segment length prior P_n. In the case of a data set with a constant sample rate and for a independence of segmentation result from the segment length, the segment length prior distribution become uniform distributed with the formulation like

$$P_n(Y_{i:j}) = \left(\frac{1}{n_{\max}}\right)^{j-i} , \tag{2.9}$$

where $n_{\max} \in \mathbb{N}^+$ is the predefined maximum segment length. In an case of an offline application, the maximum segment length is set to length of the complete observation.

Number of changepoint prior

With the prior distribution of the number of changepoints P_k, the segmentation result can be control regarding to the segment quantity. The number of changepoint prior P_k will be considered as uniform distributed, if the segmentation result has no relation to the number of changepoint. In this case the number of changepoint prior P_k is formulated like

$$P_k(k) = \left(\frac{1}{k_{\max}}\right) , \tag{2.10}$$

where k is the number of changepoints, $k_{\max}$ is the maximum number of changepoints and therefore, equal to the number of observer data points $k_{\max} = n_{\max}$.

The Segment likelihood

The introduced formulation of the segment likelihood in Eq. 2.4 is extended by the segment length prior and with the representation of a segment by the segment vector $\boldsymbol{s}_{i:j}$ of contain observation, the formulation can be written like

$$P_S(\boldsymbol{s}_{i:j}|\Theta) = \int P(\boldsymbol{s}_{i:j}|\theta)P(\theta)d\Theta P_n(\boldsymbol{s}_{i:j}) \ , \tag{2.11}$$

where the $P(\boldsymbol{s}_{i:j}|\theta)$ and $P(\theta)0$ is considered as given by the model definition.

Forward Recursion Formulation

To provide the segmentation probability of every possible segmentation as defined in Eq. (2.7), the Forward Recursion formulation define the probability of a given first changepoint $P_i(\boldsymbol{s}_{i:j}|c_1)$, where the first changepoint is placed at $c_1 \in Y_{i:j}$ and the remaining observations is segmented in any possible segmentation and can be written like

$$P(\boldsymbol{s}_{1:n}, c_1) = P_S(\boldsymbol{s}_{1:c_1})P_D(\boldsymbol{s}_{c_1+1:n}) \ , \tag{2.12}$$

where $P_D(\boldsymbol{s}_{i:j})$ is the probability of the segmented remaining observations $Y_{i:j}$. Beginning with the smallest possible remaining observations $Y_{n-1:n}$, the segmented probability can be written like

$$P_D(\boldsymbol{s}_{n-1:n}) = P_S(\boldsymbol{s}_{n-1:n}) + P_S(\boldsymbol{s}_{n-1:n})P_S(\boldsymbol{s}_{n:n}) \tag{2.13}$$

and on this base, the segmented probability of the remaining observation can be recursively formulated like

$$\begin{aligned} P_D(\boldsymbol{s}_{n-2:n}) &= P_S(\boldsymbol{s}_{n-2:n}) + P_S(\boldsymbol{s}_{n-2:n-1})P_D(\boldsymbol{s}_{n-1:n}) \\ P_D(\boldsymbol{s}_{n-2:n}) &= \sum_{i=n-1}^{n} P_S(\boldsymbol{s}_{n-2:i})P_D(\boldsymbol{s}_{i:n}) \ , \end{aligned} \tag{2.14}$$

where n is a changepoint per definition and therefore, the probability is $P_D(\boldsymbol{s}_{n:n}) = 1$. Finally, the probability of segmentation of the observation $Y_{1:n}$ is formulated like

$$P_D(\boldsymbol{s}_{1:n}) = \sum_{i=1}^{n} P_S(\boldsymbol{s}_{n-2:i})P_D(\boldsymbol{s}_{i:n}) \tag{2.15}$$

and as it contains all possible segmentations, $P_D(\boldsymbol{s}_{1:n})$ become the segmentation marginal likelihood.

Changepoint distribution

With the definition of the first changepoint in Eq. (2.12), the segmentation marginal likelihood in Eq. (2.15) and the Bayesian Theorem, the probability of the first changepoint is formulated like

$$\begin{aligned} P_c(c_1|\boldsymbol{s}_{1:n}) &= \frac{P(\boldsymbol{s}_{1:n}, c_1)}{\sum_{c_i \in C} P(\boldsymbol{s}_{1:n}|c_i)} \\ &= \frac{P_S(\boldsymbol{s}_{1:c_1}) P_D(\boldsymbol{s}_{c_1+1:n})}{P_D(\boldsymbol{s}_{1:n})} \ . \end{aligned} \tag{2.16}$$

Based on this, the probability of the following changepoints $P_c(c_k|\boldsymbol{s}_{1:n})$ can be recursively formulated like

$$P_c(c_i|c_{i-1}, \boldsymbol{s}_{1:n}) = \frac{P_S(\boldsymbol{s}_{c_{i-1}:c_i}) P_D(\boldsymbol{s}_{c_i+1:n})}{P_D(\boldsymbol{s}_{c_i:n})} \ , \tag{2.17}$$

where the number of changepoints k is limited by the number of observation n and contains always contains the first and last data point $k < n$.

Elimination of the number of changepoint dependency

The resulting changepoints density depends on the number of changepoints, by the relation to position in the changepoint sequence. To eliminate this dependency, the changepoints density marginalized and like

$$P_{cp}(y_j) = \sum_{i=0}^{k-2} P_c(c_i|y_j) P_k(i) \ , \tag{2.18}$$

where $P_k(k)$ is the prior of the number of changepoints.

2.1.5 Online Segmentation Algorithm

Similar to the offline Bayesian changepoint detection is the online Bayesian changepoint detection is based on the Product Partition model formulation. During an online application, new data points are generated in intervals, which requires an update of the segmentation result for each new data point. For this purpose, the formulation is restructured and focused on the run length r_t, which describes the time since the last changepoint to the most recent data point x_t in the observation $X_{1:t}$. Therefore, every new data point is investigated in the context of the membership to an already existing segment represented by the run lengths $r_t > 0$ or as the first data point of a new segment with run length zero $r_t = 0$. For this investigation, the growth probabilities $P(r_t = r_t + 1, Y_{1:t})$ are determined, which represent the probabilities of each possible run length increased by one, and the change point

probabilities $P(r_t = 0, Y_{1:t})$ as the probabilities when the run lengths become zero. Here, the posterior predictive distribution $P(y_{t+1}|Y_{1:t})$ is required.
In the next three sections, we will introduce the recursive run length posterior, the changepoint prior and the initial conditions formulations. Based on this definition the online segmentation algorithm is presented in the final section. Finally, the online segmentation algorithm is introduced.

Recursive Run Length Posterior

By integrate over the posterior distribution on the current run length, the marginal predictive distribution can be determine like

$$P(x_{t+1}|X_{1:t}) = \sum_{l=1}^{t} P(x_{t+1}|r_t = l, X_{1:l})P(r_t = l|X_{1:t}) \tag{2.19}$$

with the posterior distribution like

$$P(r_t|X_{1:t}) = \frac{P(r_t, X_{1:t})}{P(X_{1:t})} \tag{2.20}$$

. The joint distribution over the run length and the observed data in a recursive form can be written as

$$\begin{aligned} P(r_t, X_{1:t}) &= \sum_{l=1}^{t-1} P(r_t = l+1, r_{t-1} = l, X_{1:l+1}) \\ &= \sum_{l=1}^{t-1} P(r_t = l+1, x_{l+1}|r_{t-1} = l, X_{1:l})P(r_{t-1} = l, X_{1:l}) \\ &= \sum_{l=1}^{t-1} P(r_t = l+1|r_{t-1} = l)P(x_l|r_{t-1} = l, X_{1:l})P(r_{t-1} = l, X_{1:l}) \end{aligned} \tag{2.21}$$

, where the conditional prior $P(r_t|r_{t-1})$ can be interpreted as the changepoint prior and determine the transition of the run length. The predictive distribution $P(x_l|r_{t-1} = l, X_{1:l})$ is based on the underlying probabilistic model, which provide a predictive probability of the new data point based on the data since the last changepoint.

Posterior Predictive in Underlying Probabilistic Model

By the assumption of an underlying probabilistic model based on the exponential family and a conjugate prior, the posterior predictive can be formulated by on the observation like

$$P(x_{j+1}|X_{i:j}) = \int P(x_{j+1}|\theta)P(\theta|X_{i:j})d\theta \tag{2.22}$$

, where the hyperparameter θ are defined by the exponential distribution and the chosen conjugate prior $P(\theta|X_{i:j})$. In the context of the run length the posterior predictive can be formulated like

$$P(x_{j+1}|r_t, X_{i:j}) = \int P(x_{j+1}|\theta)P(\theta|r_t, X_{i:j})d\theta \tag{2.23}$$

, where the conjugate prior is based on the observation, which is limited by the run length.

Changepoint prior

Based on the binary of the possible transition, the changepoint prior have only two possible outcome, where a changepoint will result in $r_t = 0$ and the no changepoint extend the run length by one $r_t = r_{t-1} + 1$. A definition of the changepoint prior can be formulated like

$$P(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}+1) & \text{if } r_t = 0 \\ 1 - H(r_{t-1}+1) & \text{if } r_t = r_t + 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

, where $H(\tau)$ is the hazard function and in the case of a discrete geometric distribution can be formulated like $H(\tau) = \frac{1}{\lambda}$ with the as a constant timescale λ.

Boundary Conditions

For the initial condition, the first data point of the observation can be interpreted as the first point after a changepoint. In this case the run length prior is defined by $P(r_0 = 1)$.

Segmentation Algorithm

A commonly used algorithm for the online Bayesian changepoint detection is formulated like:

1. **Initialize:**
   Place the underlying probabilistic model parameter on the prior value and define the the initial condition, where we assume a changepoint at $y_{t=0}$. The initial definitions are formulated like

$$\begin{aligned} \theta_t &= \theta_0 \\ P(r_0 = 0) &= 1 \end{aligned} \tag{2.25}$$

   , where the underlying probabilistic parameter vector $\theta_t \in \mathbb{R}^t$ depend on the run length and the θ_0 describe the prior values.

2. **Received New Data Point** y_t

3. **Predictive Probabilities**
The predictive probabilities of the new data point for each possible run lengths $\{1 \ldots r_{t-1}\}$ are computed like:

$$\tau_t = P(x_t|r_{t-1}, X_{1:t-1}) \quad = \int P(x_t|\theta)P(\theta|r_t, X_{1:t-1})d\theta \tag{2.26}$$

, where the predictive probabilities vector $\tau \in \mathbb{R}^t$ is defined to structure the calculation.

4. **Growth Probabilities**
The probability of the next incremental increased run length r_t depends only on the previous run lengthr_{t-1} and therefore, the growth probability is formulated like:

$$P(r_t = r_{t-1} + 1, Y_{1:t}) = P(r_{t-1}, X_{1:t-1})P(x_t|r_{t-1}, X_{1:j})(1 - H(r_{t-1})) \tag{2.27}$$

5. **Changepoint Probabilities**
The probability of a changepoint at the new location depends on the probabilities of all previous run lengths $\{1 \ldots r_{t-1}\}$ and therefore, the changepoint probability is formulated like:

$$\begin{aligned} P(r_t = 0, X_{i:t}) &= \sum_{l=1}^{r_{t-1}} P(r_l, X_{1:t-1})P(x_{j+1}|r_t, X_{i:j})H(r_{t-1}) \\ &= \sum_{l=1}^{t-1} P(r_t = l + 1|r_{t-1} = l)P(x_l|r_{t-1} = l, X_{1:l})P(r_{t-1} = l, X_{1:l}) \end{aligned} \tag{2.28}$$

6. **Evidence**
The observation extended by the new data point, results in the evidence update like:

$$P(Y_{1:t}) = \sum_{l=1}^{t} P(r_t = l, Y_{1:t}) \tag{2.29}$$

7. **Run Length Distribution**
The run length posterior will be update like:

$$P(r_t|Y_{1:t}) = \frac{P(r_t, Y_{1:t})}{P(Y_{1:t})} \tag{2.30}$$

8. **Underlying Probabilistic model parameter**
The underlying probabilistic model parameter vector is updated for all new run lengths like

$$\theta_t = g(\theta_{t-1}, x_t) \tag{2.31}$$

, where the update function $g(\theta, x)$ depends on the formulated conjugate prior and exponential distribution of the underlying probabilistic model.

9. **Prediction**
A further prediction for a the data point y_{t+1} can be computed like

$$P(y_{t+1}|Y_{1:t}) = \sum_{l=1}^{t} P(y_{t+1}|r_t = l, Y_{1:l})P(r_t = l|Y_{1:t}) \tag{2.32}$$

10. **Next Data Point**
Finally, the index is moved like t to $t-1$ and it continues with the return to Step 2.

2.1.6 Computational Effort Analysis

As the number of segments increases quadratically with the number of observed data points $O(n^2)$, the computational effort of the BCPD algorithm becomes prohibitive longer observations. To improve its efficiency, we will introduce the logarithm formulation of the segmentation problem and the pruning condition formulated by Paul Fearnhead [Fea06].

Logarithm formulation

The logarithm formulation allows as well as the maximum likelihood estimation to simply and accelerate the probability computations. By replacing the likelihood with the log likelihood of the normal distribution, the computation of the exponential can be avoided. Additionally, the numerically stability of the disappearing small joint probabilities values for longer segments can be replaced by the sum of log likelihoods, which numerically stabilizes computations. The logarithm formulation of the segment probability density can be written as

$$L_S(\boldsymbol{s}_{i:j}|\Theta) = \sum_{\theta\in\Theta} \sum_{l=i}^{j} L_M\left(y_l|x_l, \theta)\right) L(\theta|\Theta) \ , \tag{2.33}$$

where the notation L indicates the logarithm formulation version of the segment density.

Pruning Condition

with a progressive computation of the segment likelihoods by increment segment length, the segment likelihood trend can be observed. If the observed segment exceed one or more changepoints, the segment will become unlikely and the likelihood will drop down of a low level. To avoid following unnecessary computations of unlikely segments, Paul Fearnhead introduced in his paper [Fea06] a truncating conditions, which allows to prune the segment likelihood computations with negligible error. The prune condition is defined like

$$\eta_{\text{prune}} < \frac{P_s(\boldsymbol{s}_{i:j})P_D(\boldsymbol{s}_{i:j})}{\sum_{s=i}^{j} P_s(\boldsymbol{s}_{i:s})P_D(\boldsymbol{s}_{s:j})} \ , \tag{2.34}$$

where the pruning threshold $\eta_{\text{prune}} \in \mathbb{R}$ is a predetermined value. With an observation, where the number of changepoints increases roughly linear with the number of observations n, Paul Fearnhead predicts the computational effort as reduced to linear increasing $O(n)$.

2.2 Robot Control

In this section, we will introduce the formulation of the robot dynamic and robot control architecture, which enables the kinaesthetic teaching.

2.2.1 Robot Dynamics

The robot dynamics provide the relationship between actuation and contact forces and the resulting acceleration and motion trajectories [SK16](p.36). In this thesis we consider the robot dynamical model based on the Euler-Lagrange equations in Matrix form [SK16] for an n flexible joints [AOFH03] as

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) &= \boldsymbol{K}(\theta - \boldsymbol{q}) + \boldsymbol{\tau}_{\text{ext}} \\ \boldsymbol{\tau}_m &= \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q}) + \boldsymbol{B}\ddot{\boldsymbol{\theta}} \ , \end{aligned} \tag{2.35}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^n$ are the joint angels, joint twists and joint accelerations, $\boldsymbol{\theta} \in \mathbb{R}^n$ is the motor position vector and $\mathbb{R}^n$ is the joint space . $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^n$ and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ are defined as the inertia matrix, centripetal and Coriolis vector and the gravity vector, respectively. The joint stiffness is described by the diagonal matrix $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ and the motor inertia is defined by diagonal matrix $\boldsymbol{B} \in \mathbb{R}^{n\times n}$. The external acting torque is described by the external torque vector $\boldsymbol{\tau}_{\text{ext}}$, the joint torque vector results from $\boldsymbol{\tau}_{\text{j}} = \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q})$ and $\boldsymbol{\tau}_{\text{m}}$ is the motor torque vector, which is used as control input. To control the robot with respect to the desired torque $\boldsymbol{\tau}_d$, the state feedback controller defined in [AOFH03] with is formulation:

$$\boldsymbol{\tau}_m = \boldsymbol{\tau}_d - \boldsymbol{K}_T(\boldsymbol{\tau} - \boldsymbol{\tau}_d) - \boldsymbol{K}_S\dot{\boldsymbol{\tau}} \tag{2.36}$$

is used. The positive-definite controller matrices $\boldsymbol{K}_T$ and $\boldsymbol{K}_S$ stabilize the torque dynamics equilibrium: $\boldsymbol{\tau} = \boldsymbol{\tau}_d$. With the singular perturbation consideration the robot model and link dynamics result in:

$$\bar{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}_d + \boldsymbol{\tau}_{\text{ext}} \tag{2.37}$$

with $\bar{\boldsymbol{M}}(\boldsymbol{q}) = (\boldsymbol{M}(\boldsymbol{q}) + (\boldsymbol{I} + \boldsymbol{K}_T)^{-1}\boldsymbol{B})$.

Operational Space - Cartesian space

The conversion of the robot model defined in the joint space into a definition in the cartesian space allows the aligning of the control goals with the task goals and manipulators in a common space, which leads to the term operational space. This transformation of the robot structure of joints and links is known as the forward kinematic. The function of forward kinematic describes the position and orientation vector $\boldsymbol{x} \in \mathbb{R}^6$ of the end effector based on the joint angles in combination with the robot model. This function is written as $\boldsymbol{x} = f_{\text{f.k.}}(\boldsymbol{q})$. The Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}) = \frac{\delta f_{\text{f.k.}}(\boldsymbol{q})}{\delta \boldsymbol{q}}$ provides a transformation of the joint angle dynamics $(\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}})$ into the end effector velocity vector $\dot{\boldsymbol{x}} \in \mathbb{R}^6$ and the acceleration $\ddot{\boldsymbol{x}} \in \mathbb{R}^6$ vector as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{x}} &= \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \ . \end{aligned} \tag{2.38}$$

In the scope of this thesis, we assume the Jacobian $\boldsymbol{J}(\boldsymbol{q})$ to have full row rank is considered workspace, which means the singularity treatment does not have to be taken into account. The robot model in cartesian space can be obtained by the substitution of Eq. (2.38) into Eq. (2.37) and leads into the equation:

$$\boldsymbol{f}_{\text{ext}} + \boldsymbol{f}_c = \boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{x}} + \boldsymbol{\eta}(\boldsymbol{q}) \ , \tag{2.39}$$

where $\boldsymbol{f}_{\text{ext}} \in \mathbb{R}^6$ and $\boldsymbol{f}_c \in \mathbb{R}^6$ denotes the external and the command forces in Cartesian space. The relation of the external wrench vector is given by $\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_{\text{ext}}$, and of the command wrench vector $\boldsymbol{\tau}_d = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_c$ The pseudo-inertia matrix $\boldsymbol{\Lambda}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ is defined like

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = \boldsymbol{J}^{-T}(\boldsymbol{q})\bar{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q}) \ . \tag{2.40}$$

The Coriolis and centrifugal effects are described by $\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{6\times 6}$ like

$$\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{J}^{-1}(\boldsymbol{q}) - \boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q}) \ . \tag{2.41}$$

Finally, the Cartesian space gravitation vector is defined as $\boldsymbol{\eta}(\boldsymbol{q}) \in \mathbb{R}^6$ and is given by

$$\boldsymbol{\eta}(q) = \boldsymbol{J}^{-T}\boldsymbol{\tau}_g(\boldsymbol{q}) \ . \tag{2.42}$$

This kinematic mapping also allows formulating robot controlling task in the cartesian space.

Cartesian Impedance Control

The goal of the Impedance control is a robot, which behaves like a spring mass damper system. For that, the desired closed-loop behavior should look like

$$\boldsymbol{\Lambda}_d\ddot{\boldsymbol{e}}_x + \boldsymbol{D}_d\dot{\boldsymbol{e}}_x + \boldsymbol{K}_d\boldsymbol{e}_x = \boldsymbol{f}_{\text{ext}} \ , \tag{2.43}$$

with the desired mass $\boldsymbol{\Lambda}_d$, the desired damping $\boldsymbol{D}_d$ and the desired stiffness matrix $\boldsymbol{K}_d$. Additionally, the position, velocity and acceleration error is defined as $\boldsymbol{e}_x = \boldsymbol{x} - \boldsymbol{x}_d$, $\dot{\boldsymbol{e}}_x = \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d$, $\ddot{\boldsymbol{e}}_x = \ddot{\boldsymbol{x}} - \ddot{\boldsymbol{x}}_d$, respectively. According to the Cartesian Impedance control formulation in [AOFH03], the control law is written as

$$\begin{aligned} \boldsymbol{f}_m &= \boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\boldsymbol{x}}_d - \boldsymbol{D}_d\dot{\boldsymbol{e}}_x - \boldsymbol{K}_d\boldsymbol{e}_x - \tilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{e}}_x - \Lambda(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{\tau}_d &= \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{f}_m + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \ , \end{aligned} \tag{2.44}$$

where $\tilde{\boldsymbol{C}}$ is used to ensure the stability of the system in free motion and in feedback interconnection with a passive environment. $\tilde{\boldsymbol{C}}$ is an arbitrary matrix, which fulfils the skew symmetry of $\dot{\boldsymbol{\Lambda}}(\boldsymbol{q}) - 2\tilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})$, for instance $\tilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}}) = 1/2\dot{\boldsymbol{\Lambda}}(\boldsymbol{q})$. Finally, the closed loop dynamics can so be formulated as

$$\boldsymbol{\Lambda}_d\ddot{\boldsymbol{e}}_x + \boldsymbol{D}_d\dot{\boldsymbol{e}}_x + \boldsymbol{K}_d\boldsymbol{e}_x + \tilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{e}}_x = \boldsymbol{f}_{\text{ext}} \ , \tag{2.45}$$

where this equation is used as the passive mapping from the external force $\boldsymbol{f}_{\text{ext}}$ to the velocity error $\dot{\boldsymbol{e}}_x$ [AOFH03].

Solution of the Second order Dynamic in the over damping case

A formulation of a second order dynamical model can be written as the second order differential equation as

$$a\ddot{x} + b\dot{x} + cx = d \ , \tag{2.46}$$

where the variables a, b, c and d are considered as time constants. By assuming $x = e^{rt}$ and r as the characteristic root, the homogeneous solution with $d = 0$ is written as

$$(ar^2 + br + c)e^{rt} = 0. \tag{2.47}$$

As e^{rt} will never be zero for all rt, the characteristic equation can be written as

$$ar^2 + br + c = 0 \tag{2.48}$$

and the determination of the characteristic roots r_1, r_2 is achieved by the quadratic equation formulated like

$$r_1, r_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \ . \tag{2.49}$$

In the case of a positive discriminant $b^2 - 4ac > 0$ or in the context of the spring-damper formulation better known as the over-damping case, the characteristic roots are two real values $\boldsymbol{r} \in \mathbb{R}^2$. In this case, the general solutions can now be written like

$$\begin{aligned} x(t) &= c_1e^{r_1t} + c_2e^{r_2t} + d \\ \dot{x}(t) &= c_1r_1e^{r_1t} + c_2r_2e^{r_2t} \\ \ddot{x}(t) &= c_1r_1^2e^{r_1t} + c_2r_2^2e^{r_2t} \\ &, \end{aligned} \tag{2.50}$$

where the coefficients $c_1, c_2 \in \mathbb{R}$ considered as constants and are used to satisfy the boundary conditions. Therefore, with the boundary conditions like

$$\begin{aligned} x_{t=0} &= x_0, & x_{t=\infty} &= x_d \\ \dot{x}_{t=0} &= 0, & \dot{x}_{t=\infty} &= 0 \\ &, \end{aligned} \tag{2.51}$$

where $x_d \in \mathbb{R}$ represent the desired value and by substituting the boundary conditions in Eq. (2.51)) into the general solution from Eq. (2.51)) like

$$\begin{aligned} x(t=0) &= c_1 + c_2 + d && \overset{!}{=} x_0 \\ x(t=\infty) &= d && \overset{!}{=} x_d \\ \dot{x}(t=0) &= r_1 c_1 + r_2 c_2 && \overset{!}{=} 0 \\ \dot{x}(t=\infty) &= 0 && \overset{!}{=} 0 \ , \end{aligned} \tag{2.52}$$

the coefficients c_1, c_2 and the particular variable can be determine like

$$\begin{aligned} d &= x_d \\ c_1 &= (x_0 - x_d)\left(1 - \frac{r_1}{r_1 - r_2}\right) \\ c_2 &= (x_0 - x_d)\left(\frac{r_1}{r_1 - r_2}\right) \ . \end{aligned} \tag{2.53}$$

Finally, the second order dynamical model function $h^2(x(t), x_d)$ is formulated like

$$\begin{aligned} h^2(x(t), a, b, c, x_d) : x(t) &= c_1 e^{r_1 t} + c_2 e^{r_2 t} + x_d \\ \text{with } r_1, r_2 &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad \text{and } b^2 - 4ac > 0 \ . \end{aligned} \tag{2.54}$$

Chapter 3

Informed linear models for Bayesian Changepoint Detection

To improve the segmentation efficiency of robot kinaesthetic teaching data, this chapter aims for the definition of a segment model that considers a priori knowledge about the dynamics during interaction. In Sec. 3.1, we will explore the interaction dynamics during kinaesthetic teaching in detail, including a simple human behavior model. Based on these insights, Sec. 3.2 introduces the informed segment model.

3.1 Dynamics during kinaesthetic teaching

To enable an intuitive interactive behavior, the manipulator dynamics are compensated applying the inverse dynamics [SK16]. Following the Cartesian impedance control from Sec. 2.2.1, an intuitive guiding behavior is rendered by adding a damping term, yielding

$$\boldsymbol{f}_{\text{ext}} = \boldsymbol{\Lambda}_d \ddot{\boldsymbol{x}} + \boldsymbol{K}_D \dot{\boldsymbol{x}} + \epsilon_{\text{dynamic}} \ , \tag{3.1}$$

where $\epsilon_{\text{dynamic}}$ represents model errors.
External forces result from both the human and the environment, i.e.

$$\boldsymbol{f}_{\text{ext}} = \boldsymbol{f}_{\text{human}} + \boldsymbol{f}_{\text{env}} \ , \tag{3.2}$$

where $\boldsymbol{f}_{\text{human}}$ is the human exerted force, $\boldsymbol{f}_{\text{env}}$ is the environmental force vector. See Fig. 3.1 for an illustrative representation.

3.1.1 Human model

With the human considered as an operator of an outer loop control, the intention goals become the control inputs and the human haptical, and the visual sensing allows to trace back the control output. The control structure is illustrated in Fig. 3.2, where the diversities in the human sensing are move into the control function.

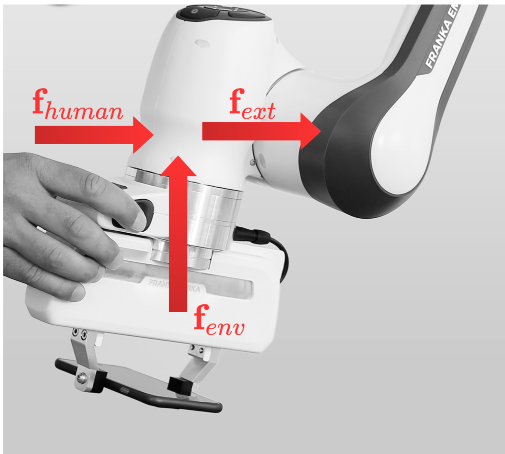


Figure 3.1: This figure shows the resulting force observed during kinaesthetic teaching.

Due to the combination of the complexity of the eye-hand coordination and the diversity in the execution process of each individual human teacher, the human-generated force signal will differ from any known control signal behaviors. In terms of intention recognition, the determination of the desired position and the desired force as the goal of the motion and the exerted force is paramount. So, we designed a model based on a proportional controller, to track intention with a possible intention goal and added a the probabilistic error term as the compensation of discrepancies like $\epsilon_{\text{human}} \propto \mathcal{N}(0, \sigma^2_{\text{human}})$.

The contact model is used to model various interactions between the robot and its environment. Since the modeling of a contact process relies on detailed information about the environment, which is excluded in the approach, we define discrete states to model the contact model, where deviations during the transitions are compensated by model uncertainties.

Within the scope of our work we have focused on the three following contact states, whereby the defined contact variable u is used to represent the current state. The first contact state $u = 1$ represents the robot in the open space, where no contact with the environment is assumed. In the second contact state $u = 2$ we assume a solid contact with a target object. The third contact state $u = 3$ is the transport of a target object. Therefore the contact variable u defined like

$$u = \begin{cases} 1 \text{ , in free space} \\ 2 \text{ , in contact} \\ 3 \text{ , during the transporting} \end{cases} \tag{3.3}$$

and with the decision vector $\boldsymbol{u}(u)$ mapped by the contact variable u like

$$\boldsymbol{u}(u) = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, & \mathrm{u} = 1 \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, & \mathrm{u} = 2 \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, & \mathrm{u} = 3 \end{cases} , \tag{3.4}$$

the environmental force is formulated with the contact model like

$$\boldsymbol{f}_{\text{env}} = \boldsymbol{u} \begin{bmatrix} \epsilon_{\text{sensor}} \\ \boldsymbol{k}_{\text{object}} \boldsymbol{f}_{\text{human}} + \epsilon_{\text{object}} + \epsilon_{\text{sensor}} \\ \boldsymbol{g}m + \epsilon_{\text{load}} + \epsilon_{\text{sensor}} \end{bmatrix} , \tag{3.5}$$

where ϵ_{sensor} is the the sensor noise.
In case of contact of the robot with an object, we assume that the human force is reflected by the object with stiffness scale vector $\boldsymbol{k}_{\text{object}}$ and the uncertainty ϵ_{object}. with the assumption that the human force is only exerted in the direction of the contact point, the transmission of forces is modeled with the stiffness scale as formulated like $\boldsymbol{k}_{\text{object}} = k\boldsymbol{d}, k \in [0, 1]$, where $\boldsymbol{d}$ is the direction vector as unit vector pointed against the contact direction.
In the transport state, the environmental force is based on the object mass ($m \in \mathbb{R}, m > 0$) with the gravitational vector ($\boldsymbol{g} = \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix}$) and, whereby deviations generated by the dynamics of the object are modeled with uncertainty ϵ_{load}.
Based on presented contact model, the human control force can be formulated like

$$\boldsymbol{f}_{\text{h}}(u, \boldsymbol{f}_d, \boldsymbol{x}_d) = \boldsymbol{K}_x(\boldsymbol{x} - \boldsymbol{x}_d) + \boldsymbol{K}_f\Big(\boldsymbol{u} \begin{bmatrix} 0 \\ \boldsymbol{g}m \\ \boldsymbol{k}_{\text{object}} \boldsymbol{f}_{\text{h}} \end{bmatrix} - \boldsymbol{f}_d\Big) + \epsilon_{\text{human}} . \tag{3.6}$$

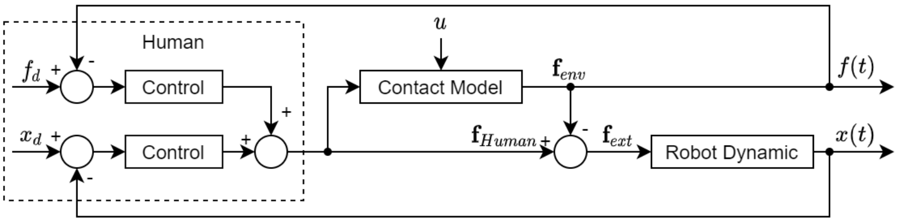


Figure 3.2: This block diagram shows a possible structure of the human as an outer loop controller.

3.2 Informed Segment Model

The segment definition of the BCPD from Sec. 2.1 is based on the definition of the likelihood of the segment like $P_S(\boldsymbol{x}_{i:j})$ in Eq. (2.4). For the observation $\boldsymbol{x}$, the informed segment model is written based on the underlying model function $f(t)$ like

$$\boldsymbol{x} = f(t) + \epsilon \ , \tag{3.7}$$

where the observation is a signal the model input become the time t and the uncertainty ϵ is considered as independent, zero mean, normally distributed like $\epsilon \propto \mathcal{N}(\epsilon|0, \sigma^2)$.

To enrich the underlying model with more information, we extent the formulation of the underlying model by an linear model function $h(t, \theta)$ with the model parameter $\theta = (\beta, \mu, \sigma^2)$ like

$$h(\boldsymbol{t}, \theta) = \boldsymbol{\beta}\boldsymbol{\Phi}(\boldsymbol{t}) + \epsilon \ , \tag{3.8}$$

where $\boldsymbol{\Phi}(\boldsymbol{t})$ is the design matrix and $\boldsymbol{\beta}$ is the model coefficient vector. For a given coefficient vector of a segment, the segment likelihood of an informed segment model can be described based on the product partition model and the posterior predictive distribution.

The **posterior predictive distributions** describe the distributions of a new observation based on the already observed observations and can be written in general terms like

$$P(x|\boldsymbol{X}) = \int P(x|\theta, \boldsymbol{X}) P(\theta|\boldsymbol{X}) d\theta \ , \tag{3.9}$$

where $P(x|\theta, \boldsymbol{X})$ is the likelihood of the observation in the context of an assumed distribution and $P(\theta|\boldsymbol{X})$ is the distribution prior. The posterior predictive distribution for a given observation can thus be determined using **the marginalized likelihood** and the updated hyperparameter θ_{i+1}. By inserting the linear model function into the posterior predictive distribution, **the segment likelihood** can be written like

$$P(\boldsymbol{x}_{\boldsymbol{1:n}}|\boldsymbol{t}) = \int\int \prod_{i=0}^{n-1} P(\boldsymbol{x}_{i+1}|\boldsymbol{\beta}\boldsymbol{\Phi}(\boldsymbol{t}), \theta, \boldsymbol{x}_{0:i}) P(\beta, \theta|\boldsymbol{x}_{0:i}, \boldsymbol{t}) d\beta d\theta \ , \tag{3.10}$$

where the uncertainty ϵ is considered to be independent, zero mean, normally distributed as $\epsilon \propto \mathcal{N}(\epsilon|0, \sigma^2)$. Because the informed models are defined based on signals and therefore the observations are always in relation on the time t, the notation $|t$ is dropped.

In the three dimensional cartesian space, the combination of the observations can be written like $\boldsymbol{X}_{1:n} : [\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n}]$ and as the dimension are independent among each other, the segment likelihood from the three dimensional observations can be written like

$$P_S(\boldsymbol{X}_{1:n}) = P_S(\boldsymbol{x}_{1:n}) P_S(\boldsymbol{y}_{1:n}) P_S(\boldsymbol{z}_{1:n}) \ . \tag{3.11}$$

Finally, as the models are also dimensional independent, the segment likelihood of the x-dimension is introduced representative for all dimension and the three dimensional segment likelihood is determined by Eq. (3.11).

The online Bayesian changepoint detection algorithm formulation builds on the posterior predictive distribution and the update function of the hyperparameters to determine the run-length probability.

In summary, any informed model for segmentation based on Bayesian changepoint detection must include the definition of the posterior predictive distribution, the hyperparameter update function from the marginal likelihood and the segment likelihood.

In the following sections, the designed informed segment models for kinaesthetic instruction are presented, which take into account the dynamics Eq. (3.1), Eq. (3.2) and Eq. (3.8). In Section 3.2.1, we will first consider two models for free space movements. In the section 3.2.2 the force model is presented, which recognizes and interprets the application of forces. Section 3.2.3 introduces the contact model, which focuses on the different contact states of the robot and its environment. Finally, in Section 3.2.4 the fully integrated model is presented, which is based on the combination of movements, forces, and contact models.

3.2.1 Motion

We first consider only free space motions. In order to represent the trajectory $\boldsymbol{x}$ resulting from (3.1), in the following we present two different motion models: a linear and a second order dynamic model.

Constant Velocity Model

Inspired by the event-based heuristic segmentation technique Zero velocity Crossing (ZVC), we designed the first model based on the motion heuristics, where we assume that every motion structured into a moving and a steady part. A simple underlying model, which fulfil this approach can be labeled as constant velocity model and generate a piecewise constant robot velocity signal. Therefore, the robot position signal is considered as the integration of the generated velocity signal and the section are in the shape of a first degree of the polynomial function. The model function can be written like

$$\begin{aligned} \dot{x}(t) &= \beta_1 \\ x(t) &= \beta_1 t + \beta_0 + \epsilon \end{aligned} \tag{3.12}$$

and for an observation $\boldsymbol{x}_{1:n}$ with n data points the formulation can be written like

$$\boldsymbol{x}_{1:n} = \beta^T \boldsymbol{\Phi}(t) + \epsilon \ , \tag{3.13}$$

where the gradient value β_1 and the offset value β_0 are combined to regression coefficient the vector $\boldsymbol{\beta}$. The design matrix $\boldsymbol{\Phi}(t) \in \mathcal{R}^{n\times 2}$ is generated by setting the first column to the offset parameter like $\boldsymbol{\phi_0} = \mathbf{1}$ and the second column as the first polynominal degree like $\boldsymbol{\phi_1} = \boldsymbol{t}_{i:j}$, where $\boldsymbol{t}_{i:j}$ is the relative time vector.

Model parameter determination Based on the general regression formulation, the error term ϵ is assumed as independent, mean zero normally distributed random variable, the likelihood for a for an observation $\boldsymbol{x}_{1:n}$ with n data points can be written like

$$\begin{aligned} P(\boldsymbol{x}_{1:n}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\Phi}) &\propto N(\boldsymbol{x}_{1:n}|\boldsymbol{\beta}\boldsymbol{\Phi}, \sigma^2 \boldsymbol{I}) \\ &= \frac{1}{\sqrt{(2\pi)^2|\sigma^2\boldsymbol{I}|}} e^{-\frac{1}{2}(\boldsymbol{x}_{1:n}-\boldsymbol{\beta}\boldsymbol{\Phi})^T(\sigma^2/\kappa_0)^{-1}(\boldsymbol{x}_{1:n}-\boldsymbol{\beta}\boldsymbol{\Phi})} , \end{aligned} \tag{3.14}$$

with the observation vector $\boldsymbol{x}_{1:n}$, the design matrix $\boldsymbol{\Phi}$, the regression coefficient vector $\boldsymbol{\beta}$, the residual variance $\sigma^2 \in \mathbb{R}$, and $\boldsymbol{I}$ as the Identity matrix. The unknown probabilistic model parameter are estimated by conjugate priors. Therefore, we define the prior of regression coefficient vector $\boldsymbol{\beta}$ like

$$P(\boldsymbol{\beta}|\sigma^2) \propto N\left(\boldsymbol{\beta}|0, \frac{\sigma^2}{\kappa_0}\right) \tag{3.15}$$

and based on the zero mean of the regression formulation, the prior of model variance are defined like

$$P(\sigma^2) \propto \mathcal{X}^{-2}(\sigma^2|v_0, \sigma_0^2) , \tag{3.16}$$

where hyperparameter κ_0 is used to control in influence of the residual error and the hyperparameters v_0, σ_0^2 can be interpreted as the variance and number of virtual data points.

The marginal likelihood is based on the derivation in Sec. A.1.2 and the Bayesian theorem formulation. It can be written like

$$\begin{aligned} P(\boldsymbol{x}_{1:n}|\boldsymbol{\Phi}) &= \frac{P(\boldsymbol{x}_{1:n}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\Phi})P(\boldsymbol{\beta}, \sigma^2)}{P(\boldsymbol{\beta}, \sigma^2|\boldsymbol{x}_{1:n}, \boldsymbol{\Phi})} \\ &= \frac{Z_{N\mathcal{X}^{-2}}(\mu_n, \kappa_n, v_n, \sigma_n^2)}{Z_{N\mathcal{X}^{-2}}(\mu_0, \kappa_0, v_0, \sigma_0^2)} \frac{1}{Z_l^N} \\ &= \frac{\Gamma(v_n/2)|\kappa_n|^{-1/2}(v_0\sigma_0^2/2)^{v_0/2}}{\Gamma(v_0/2)|\kappa_0|^{-m/2}(v_n\sigma_n^2/2)^{v_n/2}} \frac{1}{(2\pi)^{n/2}} \\ &= \frac{\Gamma(v_n/2)|\kappa_n|^{-1/2}(v_0\sigma_0^2)^{v_0/2}}{\Gamma(v_0/2)|\kappa_0|^{-m/2}(v_n\sigma_n^2)^{v_n/2}} \frac{1}{(\pi)^{n/2}} \end{aligned} \tag{3.17}$$

with the hyperparameter updates written like

$$\begin{aligned} \kappa_n &= \boldsymbol{\Phi}^T\boldsymbol{\Phi} + \kappa_0 I \\ v_n &= v_0 + n \\ s_n &= v_0\sigma_0^2 + \kappa_0\boldsymbol{\beta}^{\star T}\boldsymbol{\beta}^{\star} + (\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^{\star}\boldsymbol{\Phi})^T(\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^{\star}\boldsymbol{\Phi}) \\ \sigma_n^2 &= \frac{s_n}{v_n} . \end{aligned} \tag{3.18}$$

The posterior predictive distribution can be written by comparing the formulation the standard t-distribution formulated from Eq. (A.16) and the posterior predictive derived in Eq. (A.10) like

$$P(x_j|\boldsymbol{x}_{1:n}, \boldsymbol{\Phi}, \phi_j) = t_v\left(x_j \middle| \boldsymbol{\beta}^\star \boldsymbol{\phi}, \frac{(\kappa_n+1)\sigma_n^2}{\kappa_n}\right) . \tag{3.19}$$

The segment likelihood is formulated like

$$P(\boldsymbol{x}_{1:n}) = \prod_{i=0}^{n-1} t_{v_i}\left(x_{i+1} \middle| \boldsymbol{\beta}^\star \boldsymbol{\phi}, \frac{(\kappa_i+1)\sigma_i^2}{\kappa_i}\right) \tag{3.20}$$

and the segment log likelihood formulation like

$$L(\boldsymbol{x}_{1:n}) = \sum_{i=0}^{n-1} \ln\left(t_{v_i}\left(x_{i+1} \middle| \boldsymbol{\beta}^\star \boldsymbol{\phi}, \frac{(\kappa_i+1)\sigma_i^2}{\kappa_i}\right)\right) . \tag{3.21}$$

Second Order Dynamic Model

By substituting the human force with the human control force model from the linear intention model from Eq. (3.6) and the robot dynamics from Eq. (3.1) in force Eq. (3.2) and by assuming that the robot is in free space with $u = 0 \implies \boldsymbol{f}_{env} = 0$, the dynamic model is written as follows

$$\begin{aligned} \boldsymbol{K}_x(\boldsymbol{x} - \boldsymbol{x}_d) + \epsilon_{\text{human}} &= \boldsymbol{\Lambda}_d \ddot{\boldsymbol{x}} + \boldsymbol{K}_d \dot{\boldsymbol{x}} + \epsilon_{\text{dynamic}} \\ -\boldsymbol{\Lambda}_d \ddot{\boldsymbol{x}} - \boldsymbol{K}_d \dot{\boldsymbol{x}} + \boldsymbol{K}_x(\boldsymbol{x} - \boldsymbol{x}_d) &= \epsilon \ , \end{aligned} \tag{3.22}$$

where $\boldsymbol{\Lambda}_d, \boldsymbol{K}_d$ and $\boldsymbol{K}_x$ are considered as unknown controlling parameters and the uncertainties ϵ_{human} and $\epsilon_{\text{dynamic}}$ are combined in the uncertainty ϵ.
Based on the introduced general solution of second order models in the over-damping case (2.50) and with the boundary conditions formulated like

$$\begin{aligned} x(t=0) &= x_0 \quad x(t=T) = x_d \\ \dot{x}(t=0) &= 0 \quad\ \dot{x}(t=T) = 0 \ , \end{aligned} \tag{3.23}$$

the second order dynamical model function $h_{\text{SOD}}(\boldsymbol{t}, \boldsymbol{\Lambda}_d, \boldsymbol{K}_d, \boldsymbol{K}_x, \boldsymbol{x}_d)$ in one dimension can be written like

$$x(t) = c_1 e^{r_1 t} + c_2 e^{r_2 t} + x_d \ , \tag{3.24}$$

with the conditions written like

$$\begin{aligned} r_1, r_2 &= \frac{-K_d \pm \sqrt{K_d^2 - 4\Lambda_d K_x}}{2\Lambda_d}, \qquad K_d^2 - 4\Lambda_d K_x > 0, \\ c_1 &= (x_0 - x_d)\left(\frac{r_1}{r_2 - r_1} - 1\right) \quad \text{and } c_2 = (x_0 - x_d)\left(\frac{-r_1}{r_2 - r_1}\right) . \end{aligned} \tag{3.25}$$

Model Parameter Determination As the model parameters are partly constrained, a closed form solution is not available. As an alternative, we aim for an approximated solution that relies on a set of precomputed profiles.
By substituting the coefficient c_1 and c_2 as defined in (3.25) into the position function (3.24) and by moving the formulation to the start boundary condition, the position function can written like

$$\begin{aligned} x(t) &= (x_d - x_0)\left(\frac{r_1}{r_1 - r_2}\right)e^{r_1 t} + (x_d - x_0)\left(1 - \frac{r_1}{r_1 - r_2}\right)e^{r_2 t} + x_d \\ x(t) &= (x_d - x_0)\underbrace{\left[\left(\frac{r_1}{r_1 - r_2}\right)e^{r_1 t} + \left(1 - \frac{r_1}{r_1 - r_2}\right)e^{r_2 t}\right]}_{e(t)} + x_d \ , \end{aligned} \tag{3.26}$$

where the formulation can be interpreted as linear function with the design error function $e(t)$ and the start position x_0 as the offset. Additionally, the model parameter become $\theta : (x_0, x_d, r_1, r_2)$, and the model parameter unconstrained model parameter x_0 and x_d are separated from r_1, r_2, which are constrained like formulated in (3.25) and conditioned by $\boldsymbol{K}_x$, $\boldsymbol{K}_d$ and $\boldsymbol{\Lambda}_d$
As every motion starts at the first point of the observation and ends at the last point, the start position and desired position are considered like

$$x_0 = x(0) \quad x_d = x(T) \ . \tag{3.27}$$

To limit the control parameter to the over-damping case and to control values with realistic motion generation, the control parameter search space is predefined as the set of possible control parameter M.
By combining all possible control parameter, a set the design error functions $e_i(t)$ is generated and collected as a library in the design matrix $\Phi(t) \in \mathbb{R}^{m \times n}$.
Finally, the best matching set of control parameter can be determined,by minimizing of the sum-of-squared residual error function as

$$\beta^\star = \underbrace{\arg\min}_{\beta} \sum \left(\frac{x_i - x_d}{x_0 - x_d} - \beta\boldsymbol{\Phi}(t)\right)^2 \ , \tag{3.28}$$

where β is the decision vector and therefore, a binary vector with one entry set to one and the remaining to zero.
With the determined second order dynamic model parameter $\beta^\star$ by Eq. (3.28) and with the simulated model position vector $\boldsymbol{x}'_{1:n} = \boldsymbol{\beta}^\star\boldsymbol{\Phi}_{1:n}$, the probability of the segment vector $\boldsymbol{x}_{1:n}$ can be written like

$$P_S(\boldsymbol{x}_{1:n}|\boldsymbol{x}'_{1:n}, \sigma^2) = \int P(\boldsymbol{x}_{1:n}|\boldsymbol{x}'_{1:n}, \sigma^2)P(\sigma^2)d\sigma^2 \ , \tag{3.29}$$

where the variance σ^2 follows a scaled inverse chi squared distribution. The prior distribution can be written like

$$P(\sigma^2) \propto \mathcal{X}^{-2}(\sigma^2|v_0, \sigma_0^2) \ , \tag{3.30}$$

where the hyperparameter v_0 can be interpreted as the virtual data points with prior variance σ_0^2 and therefore, it is used to adjust the convergence rate of the distribution variance.

The marginal likelihood is based on the derivation in Sec. A.1.3, and can be written like

$$
\begin{aligned}
p(\boldsymbol{x}_{1:n}) &= \frac{p(\boldsymbol{x}_{1:n}|\boldsymbol{x}'_{1:n}, \sigma^2)p(\sigma^2|v_0, \sigma_0^2)}{p(\sigma^2|\boldsymbol{x}_{1:n}, \boldsymbol{x}'_{1:n}, v_0, \sigma_0^2)} \\
&= \frac{\mathcal{N}(\boldsymbol{x}_{1:n}|\boldsymbol{x}'_{1:n}, \sigma^2) \times \mathcal{X}^{-2}(\sigma^2|v_0, \sigma_0^2)}{\mathcal{N}\mathcal{X}^{-2}(\boldsymbol{x}_{1:n}, \boldsymbol{x}'_{1:n}, v_n, \sigma_n^2)} \\
&= \frac{Z_{\mathcal{N}\mathcal{X}^{-2}}(v_n, \sigma_n)}{(2\pi)^{n/2} Z_{\mathcal{X}^{-2}}(v_0, \sigma_0)} \\
&= \frac{\Gamma(v_n/2)\left(\sigma_n^2 V_n/2\right)^{-v_n/2}}{\Gamma(v_0/2)\left(\sigma_0^2 v_0/2\right)^{-v_0/2}}
\end{aligned}
\tag{3.31}
$$

with the hyperparameter update written like

$$
\begin{aligned}
v_n &= v_0 + n \\
\sigma_n^2 &= \frac{1}{v_n}\left(v_0\sigma_0^2 + \sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{x}'_i)^2\right) \ .
\end{aligned}
\tag{3.32}
$$

The posterior predictive distribution is based on the derivation in Eq. (A.15) and with student t-distribution formulated like in Eq. (A.16), it is written like

$$
P(x_i|\boldsymbol{x}_{1:n}) = t_{v_n}\left(\boldsymbol{x}_{n+1}\middle|, \boldsymbol{\beta}^\star\boldsymbol{\Phi}, \sigma_n^2\right) \ . \tag{3.33}
$$

The segment likelihood can be written like

$$
P(\boldsymbol{x}_{1:n}) = \prod_{i=0}^{n} t_{v_i}\left(\boldsymbol{x}_{i+1}\middle|\boldsymbol{\beta}^\star\boldsymbol{\Phi}, \sigma_i^2\right) \tag{3.34}
$$

and the log likelihood of the segment written like

$$
L(\boldsymbol{x}_{1:n}) = \sum_{i=0}^{n} \ln\left(t_{v_i}\left(\boldsymbol{x}_{i+1}\middle|\boldsymbol{\beta}^\star\boldsymbol{\Phi}, \sigma_i^2\right)\right) \ . \tag{3.35}
$$

3.2.2 Force

To model the force, we assume the robot is in contact with the object $u = 1$ and the desired position as already reached $x_d = x$. With this assumption, the external force become $\boldsymbol{f}_{\text{ext}} = 0$ the measured environmental force can be formulated with the human control force form Eq. (3.6), the force model can written like

$$\begin{aligned} \boldsymbol{f}_{\text{env}} &= \boldsymbol{k}_f(\boldsymbol{k}_{\text{Object}}\boldsymbol{f}_{\text{env}} - \boldsymbol{f}_d) + \epsilon \\ \boldsymbol{f}_{\text{env}} &= \boldsymbol{k}(\boldsymbol{f}_{\text{env}} - \boldsymbol{f}_d) + \epsilon \ , \end{aligned} \tag{3.36}$$

where ϵ combines the uncertainties ϵ_{human} and ϵ_{object}.
In the closed loop control the force in time is described by the first order dynamic and for a desired force f_d can be written like

$$\boldsymbol{f}(t) = (\boldsymbol{f}_0 - \boldsymbol{f}_d)e^{-\boldsymbol{k}t} + \boldsymbol{f}_d \ , \tag{3.37}$$

where proportional human control coefficient is assumed like $\boldsymbol{K}_{\text{object}} = \boldsymbol{I}k, k \in \mathbb{R}$.

Model Parameter Determination

To define the process in which a force is applied to an object as one step and to represent it in one segment, we define the initial force as zero $f_0 = 0$, and the desired force as the last force of the segment $f_d = f_n \in \boldsymbol{f}_{1:n}$.
Therefore, the force can be separated like

$$\boldsymbol{f}(t) = f_d \underbrace{(1 - e^{-\boldsymbol{k}t})}_{\phi_i(t)} + \epsilon \ , \tag{3.38}$$

where $\phi_i(t)$ is the design function.
As the codomain of the exponential function ae^{-b} is limited to $[0, 1]$ with $a > 0$ and $b > 0$, the regression of the coefficient is conditioned and to ensure a stable segmentation, the design matrix $\boldsymbol{\Phi}_f$ is built from the stable of the sample vectors as row vectors based on a set of control variable K.
Thus the segment likelihood can be written like

$$P(\boldsymbol{f}_{1:n}) = \int \max_{\boldsymbol{\beta}} P(\boldsymbol{f}_{1:n}|\boldsymbol{\beta}\boldsymbol{\Phi}_f, \sigma^2)P(\sigma^2)d\sigma^2 \ , \tag{3.39}$$

where β is the decision vector and is defined like $\beta \in \{[1, 0, \dots, 0], [0, 1, \dots 0], \dots, [0, 0, \dots, 1]\}$.
The prior $P(\sigma^2)$ distribution like

$$P(\sigma^2) \propto \mathcal{X}^{-2}(\sigma^2|v_0, \sigma_0^2) \ , \tag{3.40}$$

where the hyperparameter v_0 can be interpreted as the virtual data points with prior variance σ_0^2 and therefore, it is used to adjust the convergence rate of the distribution variance.
Based on the similarity to the Second Order Dynamic Model formulation, the Force Model the segment likelihood can be determined in the same way.

The marginal likelihood is determined by substituting (3.39) and (3.40) into the derivation in Sec. A.1.3 and written like

$$\begin{aligned} P(\boldsymbol{f}_{1:n}) &= \frac{P(\boldsymbol{f}_{1:n}|\boldsymbol{f}'_{1:n}, \sigma^2)P(\sigma^2|v_0, \sigma_0^2)}{P(\sigma^2|\boldsymbol{f}_{1:n}, \boldsymbol{f}'_{1:n}, v_0, \sigma_0^2)} \\ &= \frac{\mathcal{N}(\boldsymbol{f}_{1:n}|\boldsymbol{\beta\Phi}_f, \sigma^2) \times \mathcal{X}^{-2}(\sigma^2|v_0, \sigma_0^2)}{\mathcal{N}\mathcal{X}^{-2}(\boldsymbol{f}_{1:n}, \boldsymbol{\beta\Phi}_f, v_n, \sigma_n^2)} \\ &= \frac{\Gamma(v_n/2)\,(\sigma_n^2 v_n/2)^{-v_n/2}}{\Gamma(v_0/2)\,(\sigma_0^2 v_0/2)^{-v_0/2}} , \end{aligned} \tag{3.41}$$

with hyperparameter update like

$$\begin{aligned} v_n &= v_0 + n \\ \sigma_n^2 &= \frac{1}{v_n}\Big(v_0\sigma_0^2 + \sum(\boldsymbol{x} - \boldsymbol{\beta\Phi}_f)^2\Big) . \end{aligned} \tag{3.42}$$

The posterior predictive distribution is based on the derivation in Eq. (A.15) and with student t-distribution formulated like in Eq. (A.16), it is written like

$$P(f_j|\boldsymbol{f}_{1:n}, \boldsymbol{\Phi}, \phi_j) = t_v\Big(f_j\Big|\boldsymbol{\beta}^\star\boldsymbol{\phi}, \frac{(\kappa_n+1)\sigma_n^2}{\kappa_n}\Big) . \tag{3.43}$$

The segment Likelihood is formulated like

$$P(\boldsymbol{f}_{1:n}) = \prod_{i=0}^{n-1} t_{v_i}\Big(f_{i+1}\Big|\boldsymbol{\beta}^\star\boldsymbol{\phi}, \frac{(\kappa_i+1)\sigma_i^2}{\kappa_i}\Big) \tag{3.44}$$

and the segment log likelihood formulation like

$$L(\boldsymbol{f}_{1:n}) = \sum_{i=0}^{n-1} \ln\left(t_{v_i}\Big(f_{i+1}\Big|\boldsymbol{\beta}^\star\boldsymbol{\phi}, \frac{(\kappa_i+1)\sigma_i^2}{\kappa_i}\Big)\right) . \tag{3.45}$$

3.2.3 Contact Model

By the crucial role of the contact model for the dynamics and intention models, the following Informed Segment Model is focused on the detection of contact state and generates changepoints with the switching of the contact state. By defining possible basic contact models as basic components, the segmentation result shows the task structure as sequence of the basic components. In regards to the pick and place task, we are going to design the contact models motion, transport and apply force, where during a possible post processing, higher interaction like the move to contact can be extracted.

Depending on the three contact models the expected force signal can be written like

$$\boldsymbol{f}_{\text{env}} = \begin{cases} \epsilon_{\text{sensor}} & \text{during the motion} \\ f_{\text{contact}} + \epsilon_{\text{sensor}} & \text{in contact} \\ \boldsymbol{g} m_{\text{load}} + \epsilon_{\text{load}} + \epsilon_{\text{sensor}} & \text{during the transport} \end{cases} , \tag{3.46}$$

where ϵ_{sensor} is the sensor uncertainty, f_{contact} is the expected force in contact, m_{load} is the mass of the load, $\boldsymbol{g}$ is the gravity vector and ϵ_{load} is the uncertainty generated by the load.
With the contact variable u for motion: $u : 1$, contact: $u : 2$ and transport: $u : 3$, the decision vector $\boldsymbol{u}(u)$ mapped by the contact variable u like

$$\boldsymbol{u}_i = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \ \text{u} = 1 \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \ \text{u} = 2 \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \ \text{u} = 3 \end{cases} \tag{3.47}$$

and signal distributions expected like

$$\begin{aligned} \epsilon_{\text{sensor}} &\propto \mathcal{N}(f|0, \sigma^2_{\text{sensor}}) \\ f_{\text{contact}} &\propto \mathcal{N}(f|0, \sigma^2_{\text{contact}}) \\ f_{\text{transport}} &\propto \mathcal{N}(f|m_{\text{load}}\boldsymbol{g}, \sigma^2_{\text{transport}}) \end{aligned} \tag{3.48}$$

, a formulation of the environmental force and the contact model can be written like

$$\boldsymbol{f}_{\text{env}} = \boldsymbol{u} \underbrace{\begin{bmatrix} \mathcal{N}(f|0, \sigma^2_{\text{sensor}}) \\ \mathcal{N}(f|0, \sigma^2_{\text{sensor}} + \sigma^2_{\text{contact}}) \\ \mathcal{N}(f|m_{\text{load}}\boldsymbol{g}, \sigma^2_{\text{sensor}} + \sigma^2_{\text{transport}}) \end{bmatrix}}_{\boldsymbol{\Phi}} . \tag{3.49}$$

Finally, the segment probability can be formulated like

$$P(\boldsymbol{f}_{1:n}) = \int P(\boldsymbol{f}_{1:n}|\boldsymbol{u}(u)\boldsymbol{\Phi})P(\boldsymbol{u})d\boldsymbol{u} , \tag{3.50}$$

where the contact variable u is assumed as categorical distributed. The conjugate prior written like

$$P(\boldsymbol{u}|\boldsymbol{\alpha}_0) \propto \text{Dir}(\boldsymbol{u}|\boldsymbol{\alpha}_0) , \tag{3.51}$$

where the initial distribution of $\boldsymbol{\alpha}_0$ and Dir donates the dirichlet distributed.

Model Parameter Determination

With the condition $m > 0$ and the gravity vector defined like $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$, the load load mass is determined with $m = \bar{f}_z/-g$. To ensure that the transport model contains a mass, we define the minimum mass of m_{min} and therefore, the load mass is determined like

$$m = \max\Big(m_{min}, \frac{\bar{f}_z}{-g}\Big) . \tag{3.52}$$

The marginal likelihood is written like

$$P(\boldsymbol{f}_{1:n}) = P(\boldsymbol{f}_{1:n}|\boldsymbol{u}\boldsymbol{\Phi})P(\boldsymbol{u}_n) \tag{3.53}$$

,where $P(\boldsymbol{f}_{1:n}|\boldsymbol{u}\boldsymbol{\Phi})$ $P(\boldsymbol{u_n})$ is categorical prior distribution. The categorical prior distribution is based on the classified observations $\boldsymbol{c}$ like

$$\boldsymbol{c} = \sum_{i=1}^{n} \arg\max_{\boldsymbol{u_i}} P(\boldsymbol{f}_i|\boldsymbol{u}_i\boldsymbol{\Phi}_i) \tag{3.54}$$

and is determined like

$$P(\boldsymbol{u}_n) = \frac{\boldsymbol{\alpha} + \boldsymbol{c}}{\sum_i \alpha_i + n} \ , \tag{3.55}$$

,where α_i is the initial number virtual assumed classifications for every category i.

The posterior predictive formulation is written like

$$P(f_j|\boldsymbol{f}_{1:n}) = P(\boldsymbol{f}_{1:n}|\boldsymbol{u}_n\boldsymbol{\Phi})P(\boldsymbol{u}_n) \tag{3.56}$$

with contact variable after n observations determined like

$$u_n = \arg\max_{u_n} P(\boldsymbol{f}_{1:n}|\boldsymbol{u}_n\boldsymbol{\Phi})P(\boldsymbol{u}_n) \ . \tag{3.57}$$

The segment likelihood written like

$$P(\boldsymbol{f}_{1:n}) = \prod_{i=0}^{n-1} P(f_{i+1}|\boldsymbol{u}_i\boldsymbol{\Phi}) \ , \tag{3.58}$$

where the contact variable $\boldsymbol{u}_i$ is determined with Eq. (3.54) and Eq. (3.55).

3.2.4 Combined Model

In order to detect and segment movements, contacts and force applications simultaneously, we will combine the presented Second Order Model Sec. 3.2.1 and Force Model Sec. 3.2.2 into the Contact Model Sec. 3.2.3 as part of the fully integrated model. To extend the segmentation model to the entire input space including the position and force space the model steady θ_{steady}, no-contact $\theta_{\text{no contact}}$and transport $\theta_{\text{transport}}$ is generated, where the model are formulate like

$$\begin{aligned} P(\boldsymbol{x}_{1:n}|\theta_{\text{steady}}) :& \quad P(\boldsymbol{x}|\sigma^2_{\text{steady}})P(\sigma^2_{\text{steady}}|v_0, \sigma^2_0) \\ P(\boldsymbol{f}_{1:n}|\theta_{\text{no contact}}) :& \quad P(\boldsymbol{f}|\sigma^2_{\text{sensor}})P(\sigma^2_{\text{sensor}}|v_0, \sigma^2_0) \\ P(\boldsymbol{f}_{1:n}|\theta_{\text{transport}}) :& \quad P(\boldsymbol{f}|\sigma^2_{\text{transport}})P(\sigma^2_{\text{transport}}|v_0, \sigma^2_0) \end{aligned} \tag{3.59}$$

and distributed considered like

$$\begin{aligned} P(\boldsymbol{x}_{1:n}|\theta_{\text{steady}}) \propto &\quad \mathcal{N}(\boldsymbol{x}_{1:n}|x_1, \sigma^2_{\text{steady}})\mathcal{X}^{-2}(\sigma^2_{\text{steady}}|v_0, \sigma^2_0) \\ P(\boldsymbol{f}_{1:n}|\theta_{\text{no contact}}) \propto &\quad \mathcal{N}(\boldsymbol{f}_{1:n}|0, \sigma^2_{\text{sensor}})\mathcal{X}^{-2}(\sigma^2_{\text{sensor}}|v_0, \sigma^2_0) \\ P(\boldsymbol{f}_{1:n}|\theta_{\text{transport}}) \propto &\quad \mathcal{N}(\boldsymbol{f}_{1:n}|\boldsymbol{g}m_{\text{load}}, \sigma^2_{\text{transport}})\mathcal{X}^{-2}(\sigma^2_{\text{transport}}|v_0, \sigma^2_0)\ , \end{aligned} \tag{3.60}$$

where $\boldsymbol{g}$ is the gravity vector, m is the object mass.
The $P(\boldsymbol{x}|\theta_{\text{motion}})$ is defined similar to the model definition of the second order dynamic model and the $P(\boldsymbol{x}|\theta_{\text{contact}})$ is defined similar to model definition in of the force model, where θ_{motion} and θ_{contact} collect the respective model parameters.
The contact model are generated with the contact variable $\boldsymbol{u}$ and formulated like

$$P(\boldsymbol{x}_{1:n}, \boldsymbol{f}_{1:n}) = \boldsymbol{u}\begin{bmatrix} P(\boldsymbol{x}_{1:n}|\theta_{\text{motion}})P(\boldsymbol{f}_{1:n}|\theta_{\text{no contact}}) \\ P(\boldsymbol{x}_{1:n}|\theta_{\text{steady}})P(\boldsymbol{f}_{1:n}|\theta_{\text{contact}}) \\ P(\boldsymbol{x}_{1:n}|\theta_{\text{motion}})P(\boldsymbol{f}_{1:n}|\theta_{\text{transport}}) \end{bmatrix}. \tag{3.61}$$

Model Parameter Determination

The model parameters θ_{motion} and θ_{contact} are determined similar model parameter determination introduced in the respective models. The object mass m_{load} is determined in Eq. (3.52).

The marginal likelihoods for each model is formulated like

$$P(\boldsymbol{x}_{1:n}) = \int P(\boldsymbol{x}_{1:n}|\theta)P(\theta)d\theta\ , \tag{3.62}$$

where θ is individual model parameter and the formulation with the force as input space results by replacing the $\boldsymbol{x}_{1:n}$ with $\boldsymbol{f}_{1:n}$.

The posterior predictive is determined on the basis of Eq. (3.33), and Eq. (3.43) and can be written like

$$P(x_i, f_i|\boldsymbol{x}_{1:n}, \boldsymbol{f}_{1:n}) = \max_{\boldsymbol{u}} \boldsymbol{u}\begin{bmatrix} P(x_i|\boldsymbol{x}_{1:n}, \theta_{\text{motion}})P(f_i|\boldsymbol{f}_{1:n}, \theta_{\text{no contact}}) \\ P(x_i|\boldsymbol{x}_{1:n}, \theta_{\text{steady}})P(f_i|\boldsymbol{f}_{1:n}, \theta_{\text{contact}}) \\ P(x_i|\boldsymbol{x}_{1:n}, \theta_{\text{motion}})P(f_i|\boldsymbol{f}_{1:n}, \theta_{\text{transport}}) \end{bmatrix}. \tag{3.63}$$

The segment likelihood is formulated like

$$P(\boldsymbol{x}_{1:n}, \boldsymbol{f}_{1:n}) = \max_{\boldsymbol{u}} \boldsymbol{u}\begin{bmatrix} P_{\text{motion}}(\boldsymbol{x}_{1:n})P_{\text{no contact}}(\boldsymbol{f}_{1:n}) \\ P_{\text{steady}}(\boldsymbol{x}_{1:n})P_{\text{contact}}(\boldsymbol{f}_{1:n}) \\ P_{\text{motion}}(\boldsymbol{x}_{1:n})P_{\text{transport}}(\boldsymbol{f}_{1:n}) \end{bmatrix} \tag{3.64}$$

with the segment likelihood of each model formulated like

$$
\begin{aligned}
P_{\text{steady}}(\boldsymbol{x}_{1:n}) &= \prod_{i=0}^{n-1} t_{v_i}(x_{i+1}|x_0, \sigma^2_{\text{steady},i}) \ , \\
P_{\text{no contact}}(\boldsymbol{f}_{1:n}) &= \prod_{i=0}^{n-1} t_{v_0}(f_{i+1}|0, \sigma^2_{\text{not contact},i}) \ , \\
P_{\text{transport}}(\boldsymbol{f}_{1:n}) &= \prod_{i=0}^{n-1} t_{v_0}(f_{i+1}|\boldsymbol{g} m_{\text{load}}|\sigma^2_{\text{transport},i}) \ ,
\end{aligned}
\tag{3.65}
$$

P_{motion} from Eq. (3.34) and P_{contact} from Eq. (3.44).

Chapter 4

Evaluation

In this chapter, we will evaluate the presented informed segment models 3.2 with the help of following scenarios: the Three Point Motion scenario, the Parkour scenario, the Pick and Place scenario and the Soap Dispenser scenario.

4.1 Experimental Setup

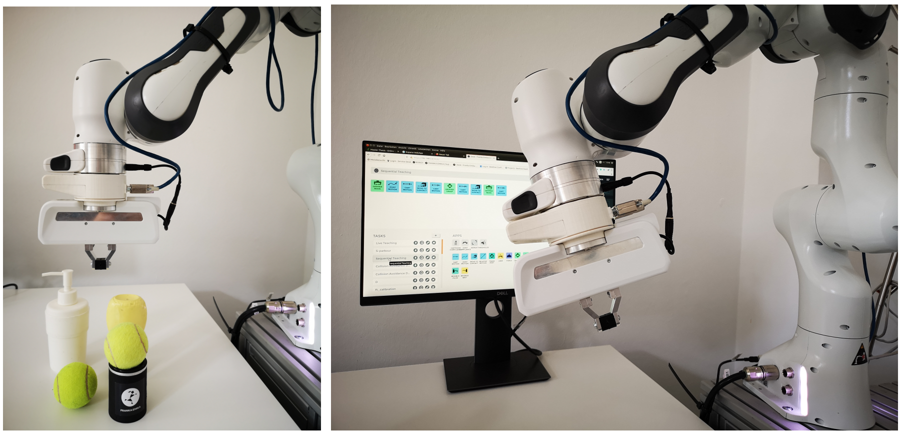


Figure 4.1: On the left side, the used objects and the used robot panda are shown. On the right side, the die robot is shown in front of the desk Graphical User interface (GUI) provided by Franka Emika.

The experimental setup is shown in Fig. 4.1 and consists of a Franka Emika panda robot [SHP20], which allows the kinaesthetic teaching by the torque-controlled joints and the Robot Control unit. To execute tasks, the panda robot brings along a GUI called Desk, which allows designing the desired task by generating a timeline, which consists of predefined apps. The robot kinematic, the wrist force, and the gripper signals are acquired during the kinaesthetic teaching by an interface, which is also provided by Franka Emika. To avoid any corruption of the force distribution by further contact points, the locations of the force impacts points are limited in a guideline for the kinaesthetic robot teaching . Therefore, the human exerted force $\boldsymbol{f}_{\text{human}}$ is limited to the last joint of the robot and therefore, above the wrist force sensor. Any kind of contact of interaction with the environment is limited to be located at the gripper jaws and therefore, on the other side of the wrist force sensor. To reduce the noise, we use the Savitziky-Golay filter to reduce the signal noise and we sampled the data with a fixed sample rate of 100hz, to achieve a uniformed time-space data set. The experimental executions and objects are based on the robot application field, which is focused on tabletop applications. The robot arm is mounted at the table and in several experimental scenarios, we will imitate real-world robot applications. In a subsequent step, the application is segmented by different segmentation models and the resulting application is executed. Based on the model parameter of every segment, we classify the task components into components labeled as move, move to contact, apply force, gripper move, gripper grasp, and transport. The classified segments are extended by the intention goals like the desired position and transformed into desk apps. Finally, the robot executes the learned task and the demonstrator evaluates if the robot executes the task as intended.

Changepoint extraction Based on the segmentation of the probability distribution for a changepoint generated by the BCPD algorithm, we extract changepoints by the threshold value $c_{\text{threshold}}$.

Grasp detection To detect the object grasps, the interface provide a binary gripper state signal. Therefore, grasping and gripper move applications are detected by inspecting the gripper state signal and is applied in a separated pipeline.

4.1.1 Experimental Scenarios

In the next section we will introduce the experimental scenarios, starting with simple robot tasks and increasing the complexity and scope of execution for each scenario.

Three-Point Motion scenario

In the first scenario a motion including three way points as illustrated in Fig. 4.2 is taught to the robot. In a real work task, this scenario can be seen as the motions between action points. The acquired robot signals are shown in the Fig. A.1.

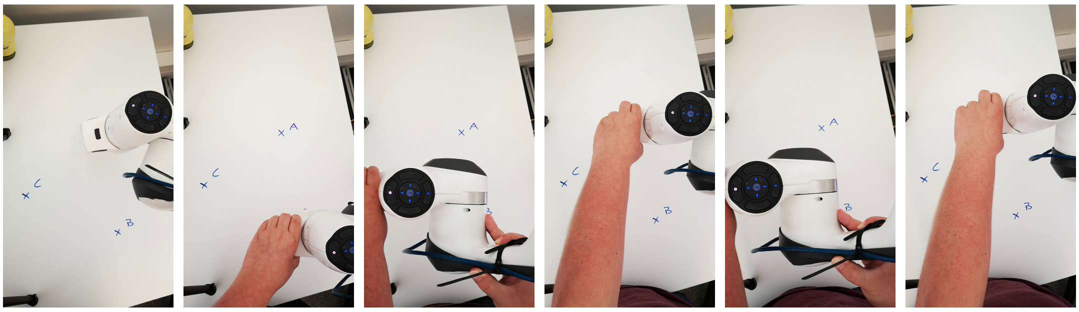


Figure 4.2: Illustration of the three point motion task execution. The robot is guided in two rounds over the three marked points (A, B, C). In the second round, point B is passed without stopping.

Parkour scenario

In this scenario, the end effector is moving through a parkour construed by two obstacles shown in Fig. 4.3. As an additional task goal, during the teaching phase and the robot execution the contact of a obstacle by the robot have to be avoided. For better tracking of object contacts, the obstacle are made by tennis balls, which start rolling way after a small impacts. This scenario simulates robot tasks in which, due to the environment, it is possible to reach the desired position by following a trajectory. The acquired robot signals are shown in the Fig. A.2.

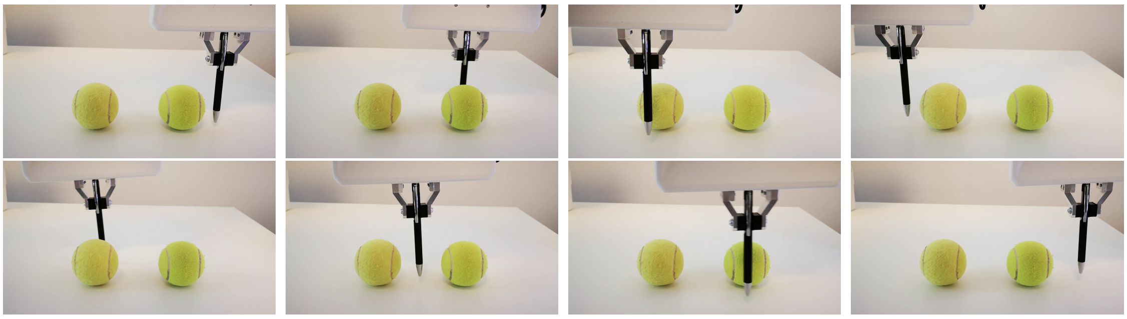


Figure 4.3: Illustration of the parkour motion task execution. The robot is guided in the form of an eight around two obstacles. Additionally, the aim of this task is not to collide with the obstacles.

Pick and Place scenario

The pick and place scenario is designed according to one of the most common robot task, where the robot is used to pick an object and place it at desired position as shown in Fig. 4.4. In general, the pick and place task can be separated into the three following sub-tasks: move to pick up position, the transport to place position and the move to end position. These task structure allows us to classify the resulting segments by assigning them to the labeled sub-tasks and therefore, the evaluation can be extended by recognition of the task structure. The acquired robot signals are shown in the Fig. A.3. The used object have a weight of 250 grams.

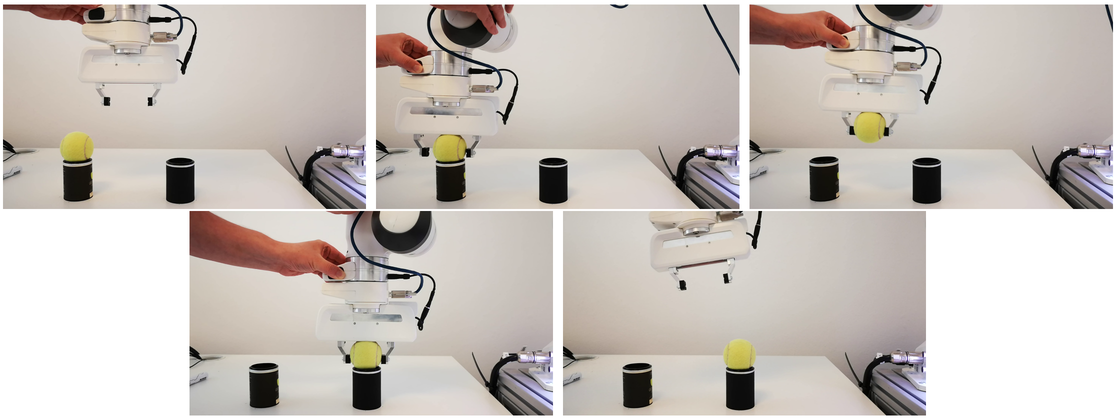


Figure 4.4: Illustration of the pick and place task execution.

Soap Dispenser

In this scenario, a commercial soap dispenser is pressed two times as shown in the Fig. 4.5. The acquired robot signals are shown in the Fig. A.4. The application has been chosen to represent tasks in which the robot is turned buttons, push buttons or push objects.

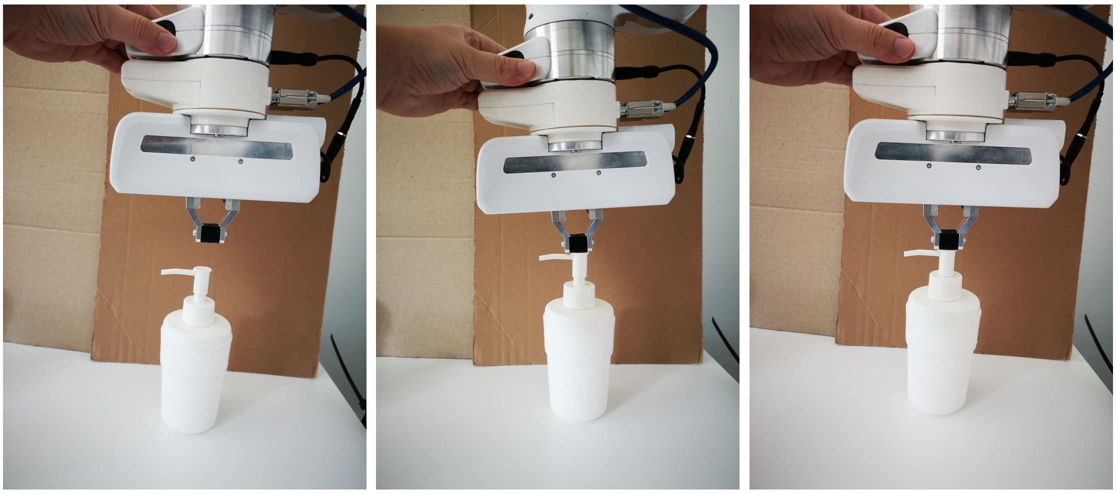


Figure 4.5: Shows left to right the start position, the first contact with the soap dispenser and finally the pressed soap dispenser with the robot.

4.2 Experimental Results

In this section, the experimental results of each introduced model and each scenario are presented and evaluated. In the first step, we will introduce the segmentation results of the introduced scenario and different models in form of resulting the probability of a changepoint, which is generated by the BCPD and the investigated probabilistic model. By extracting changepoints with an threshold filter, the segments are generated and classified by the model parameter. This sequence of task components is compared with the expected sequence of tasks, whereby missing and additional components are examined. Subsequently, the robot execution via the Desk is generated and executed. Finally, the feedback of the demonstrator is used to determine the discrepancy between the robot execution and the intended task.

4.2.1 Constant Velocity Model

The segmentation model is generated by presented formulation of the constant velocity model 3.2.1 substituted into the formulation of the BCPD algorithm 2.1 and with the parameters shown in Tab. 4.1. In the following sections we present the results of the individual scenarios in the form of a segmentation report and the evaluated feedback from the test persons regarding the intention recognition.

Configuration Parameter					
N.	f_S[hz]	$c_{\text{threshold}}$	v_0	σ_0^2	κ_0
1	10	0.2	1	0.5	1.

Table 4.1: Constant Velocity Model Configuration Parameter
This table shows configuration parameter, where N. is the parameter set number, f_s is the sample frequency in hertz, $c_{\text{threshold}}$ is the changepoint extraction threshold and the hyperparameter of the conjugate prior are v_0, σ_0^2 and κ_0

The execution of the guided task is realized by generation desk apps based on the segment parameters. As the segmentation is focused on the motion in free space, every segment is classified as motion, which is represented by the desk app cartesian motion to achieve a robot execution.

Three point motion

The segmentation report is shown in Fig. 4.6. By comparing the segmentation report with the task description, the required three points are detected, whereby additional segments like the segment $s_{2:3}$ can be classified as steady and is ignored by the subsequently execution. The summarized experimental report of the several demonstrator describe the segmentation as like intended and a correction as unnecessary.

Parkour

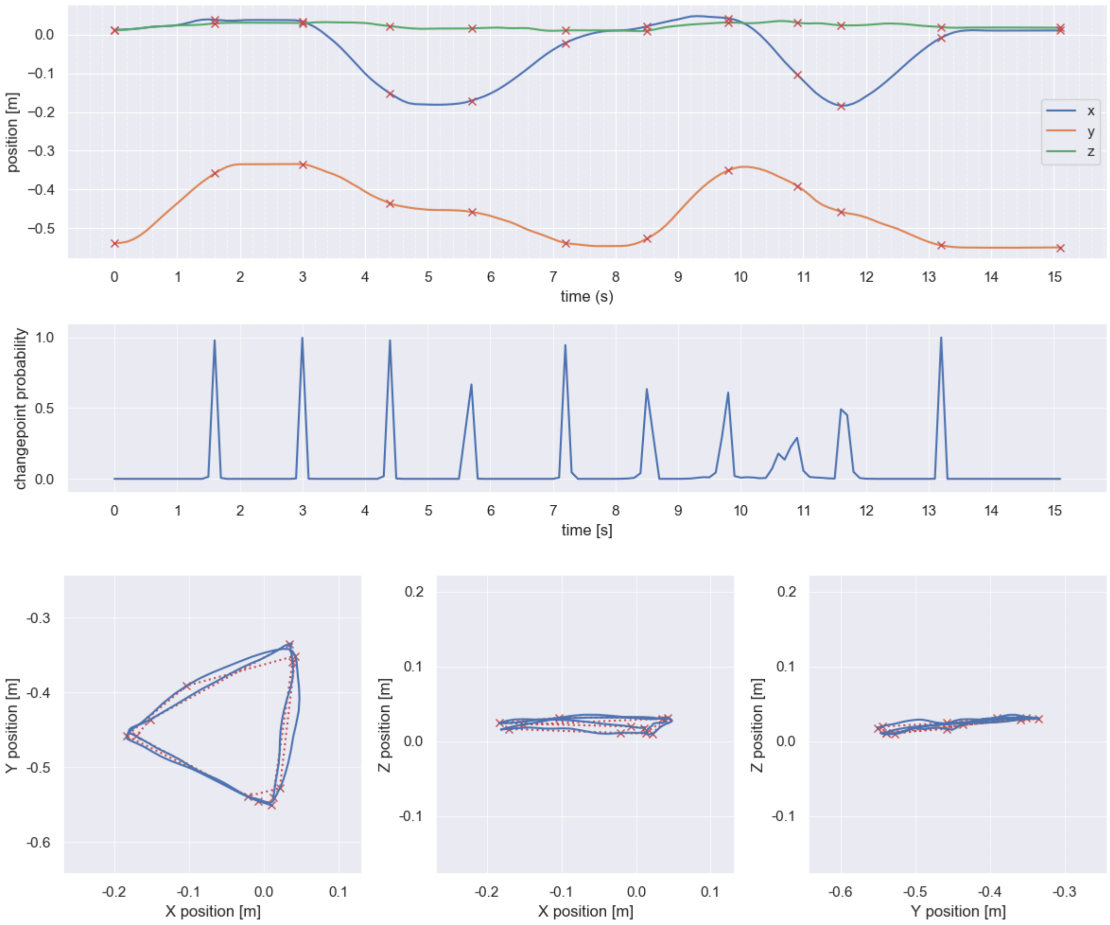
The segmentation report is shown in Fig. 4.7. The task description of this scenario is extend by the goal to avoid any collisions with the obstacles, which requires a high segmentation sensibility especially, during the passage between the two objects. The robot learned a movement without collisions. Several demonstrators reported that further corrections or an adaptive teaching process is required, whereby a movement must always be followed by a stop at the desired position. The achieve a motion without any obstacle collisions, the proband experimental reports further correction or an adaptive teaching process, where the teacher generate changepoints by a stop and go motion.

Pick and place

The segmentation report is shown in Fig. 4.8. By comparing the structure of the tasks presented, segments $s_{0:1}$, $s_{1:2}$ and $s_{2:3}$ can be allocated to the pick up movement, while segments $s_{3:4}$, $s_{4:5}$ and $s_{5:6}$ describe the transport phase. With segments $s_{6:7}$ and $s_{7:8}$ the robot is moved into the final position. Thus the segmentation result let the task structure shine through. In the experience reports of the test users the pick up point is often not reached because of a collision of a gripper and the object. Furthermore, the gripping position was set above the object, which caused the gripping of the object to fail.

Conclusion

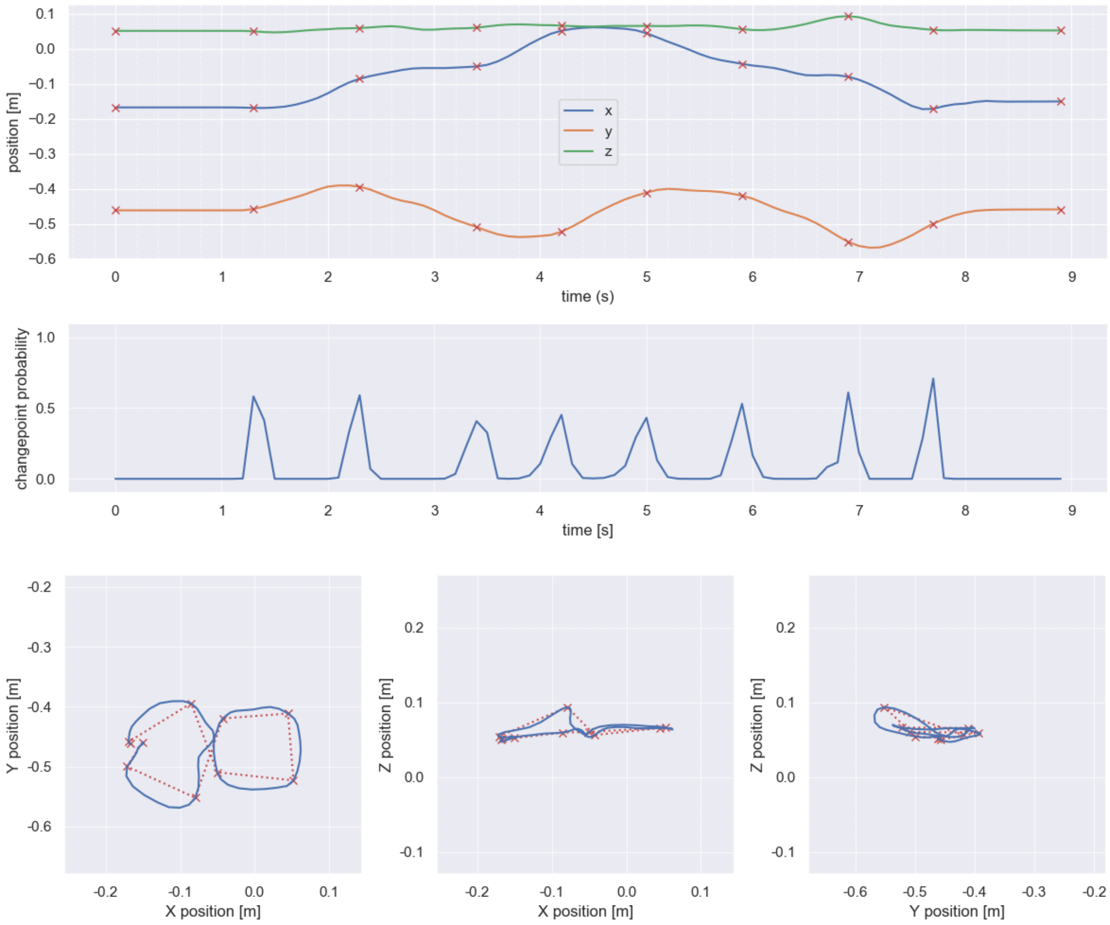
Within the definition of the model the focus was set on the velocity of the robot, which leads to the fact that the classes of the segment classification can be described as standing and moving. Thus a typical movement of the robot as described in Three-Point Scenario 4.2.1 is divided into two segments, shown in the segmentation report 4.6 by segment $s_{0:1}$ and $s_{1:2}$. The comparison with the robot motion generation of a cartesian impedance control robot and a desired position shows the segmentation of a motion into two segments is over-segmented. Generally, segmentation is described as complete, indicating that human motion profiles have a linear character



Segments Parameters

	t_0 [s]	t_n [s]	x_0 [m]			x_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	1.6	0.013	-0.54	0.012	0.039	-0.359	0.029	ConstantVelocity
$s_{1:2}$	1.6	3.0	0.039	-0.359	0.029	0.035	-0.335	0.03	ConstantVelocity
$s_{2:3}$	3.0	4.4	0.035	-0.335	0.03	-0.152	-0.436	0.022	ConstantVelocity
$s_{3:4}$	4.4	5.7	-0.152	-0.436	0.022	-0.171	-0.458	0.017	ConstantVelocity
$s_{4:5}$	5.7	7.2	-0.171	-0.458	0.017	-0.021	-0.539	0.011	ConstantVelocity
$s_{5:6}$	7.2	8.5	-0.021	-0.539	0.011	0.022	-0.528	0.01	ConstantVelocity
$s_{6:7}$	8.5	9.8	0.022	-0.528	0.01	0.043	-0.351	0.032	ConstantVelocity
$s_{7:8}$	9.8	10.9	0.043	-0.351	0.032	-0.103	-0.391	0.032	ConstantVelocity
$s_{8:9}$	10.9	11.6	-0.103	-0.391	0.032	-0.184	-0.458	0.025	ConstantVelocity
$s_{9:10}$	11.6	13.2	-0.184	-0.458	0.025	-0.008	-0.545	0.02	ConstantVelocity
$s_{10:11}$	13.2	15.1	-0.008	-0.545	0.02	0.011	-0.55	0.018	ConstantVelocity

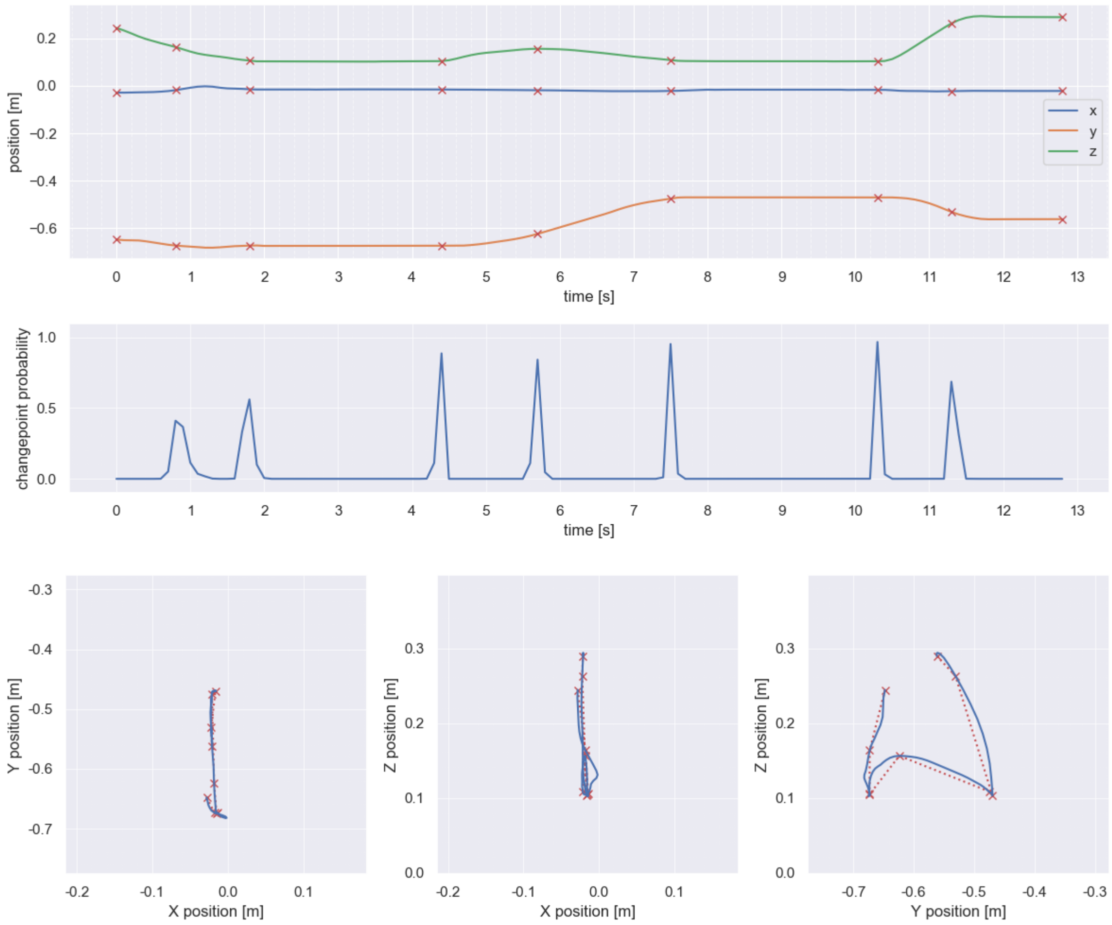
Figure 4.6: **Segmentation Report(Constant Velocity model, Three-point Motion):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The third subplot shows the motion in position space. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.



Segments Parameters

	t_0 [s]	t_n [s]	x_0 [m]			x_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	1.3	-0.167	-0.462	0.052	-0.169	-0.458	0.05	ConstantVelocity
$s_{1:2}$	1.3	2.3	-0.169	-0.458	0.05	-0.085	-0.394	0.06	ConstantVelocity
$s_{2:3}$	2.3	3.4	-0.085	-0.394	0.06	-0.05	-0.509	0.061	ConstantVelocity
$s_{3:4}$	3.4	4.2	-0.05	-0.509	0.061	0.052	-0.523	0.067	ConstantVelocity
$s_{4:5}$	4.2	5.0	0.052	-0.523	0.067	0.045	-0.411	0.066	ConstantVelocity
$s_{5:6}$	5.0	5.9	0.045	-0.411	0.066	-0.043	-0.419	0.057	ConstantVelocity
$s_{6:7}$	5.9	6.9	-0.043	-0.419	0.057	-0.08	-0.552	0.094	ConstantVelocity
$s_{7:8}$	6.9	7.7	-0.08	-0.552	0.094	-0.171	-0.5	0.055	ConstantVelocity
$s_{8:9}$	7.7	8.9	-0.171	-0.5	0.055	-0.15	-0.459	0.053	ConstantVelocity

Figure 4.7: **Segmentation Report(Constant Velocity model, Parkour):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The third subplot shows the motion in position space. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.



Segments Parameters

	t_0	t_n	x_0 [m]			x_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	0.8	-0.028	-0.648	0.244	-0.018	-0.673	0.164	ConstantVelocity
$s_{1:2}$	0.8	1.8	-0.018	-0.673	0.164	-0.014	-0.673	0.107	ConstantVelocity
$s_{2:3}$	1.8	4.4	-0.014	-0.673	0.107	-0.015	-0.674	0.104	ConstantVelocity
$s_{3:4}$	4.4	5.7	-0.015	-0.674	0.104	-0.018	-0.624	0.156	ConstantVelocity
$s_{4:5}$	5.7	7.5	-0.018	-0.624	0.156	-0.021	-0.475	0.109	ConstantVelocity
$s_{5:6}$	7.5	10.3	-0.021	-0.475	0.109	-0.017	-0.47	0.104	ConstantVelocity
$s_{6:7}$	10.3	11.3	-0.017	-0.47	0.104	-0.022	-0.531	0.263	ConstantVelocity
$s_{7:8}$	11.3	12.8	-0.022	-0.531	0.263	-0.021	-0.562	0.29	ConstantVelocity

Figure 4.8: **Segmentation Report(Constant Velocity model, Pick and Place):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The third subplot shows the motion in position space. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.

4.2.2 Second Order Dynamic Model

The second order dynamic segmentation model is generated by substituting the second order dynamic model formulation 3.2.1 into the formulation of the BCPD algorithm 2.1. Additionally, the Table 4.2 shows the used configuration parameter sets. In the following sections we present the results of the individual scenarios in the form of a segmentation report and the evaluated feedback from the test persons regarding the intention recognition.

Configuration Parameter Sets					
N.	f_S[hz]	$c_{\text{threshold}}$	(v_0, σ_0^2)	K	D
1	10	0.2	$(1, 0.05^2)$	$\{4, 8, 12, 16, 20, 24\}$	$\{10, 32.5, 55, 77.5, 100\}$
2	10	0.2	$(1, 0.025^2)$	$\{4, 8, 12, 16, 20, 24\}$	$\{10, 32.5, 55, 77.5, 100\}$

Table 4.2: Configuration Parameter Sets
This table shows the used configuration parameter sets. The column $N.$ is used to index the parameter sets and possible units are defined in the brackets like [unit]. The column f_s describe the sample frequency, v_0 and σ_0^2 are the hyperparameter of the conjugate prior ,and the stiffness and damping parameter sets are labeled by K and D.

Based on the segmentation result, the robot should then perform the task. To enable this, the segments are translated into desk apps. Since these are only movements, the Cartesian Motion app is used and parameterized with a default velocity, acceleration and the desired position. Finally, based on the feedback and necessary corrections, the robot execution can be evaluated in the context of the intention recognition.

Three point motion

The resulting segmentation for the first configuration set is shown in the Fig. 4.9. Points A, B and C marked in the task description are identified in the segmentation results by changepoints and the resulting segments result in movements to desired positions. So the robot reaches the marked point B in segments $s_{0:1}$ and $s_{3:4}$, the marked point C in segments $s_{1:2}$ and $s_{4:5}$ and the marked point A in segments $s_{3:4}$ and $s_{5:6}$. For point C the segmentation result shows an error, which places the changepoint offset. Between the desired positions for point C and point C there is a deviation in position space.
All testing demonstrator described the robot execution as like desired and experienced a correction of the execution as unnecessary.

Parkour

An example segmentation result of the parkour task with the first configuration set is shown in the Fig. 4.10. In the frame of this task, the object collision avoidance

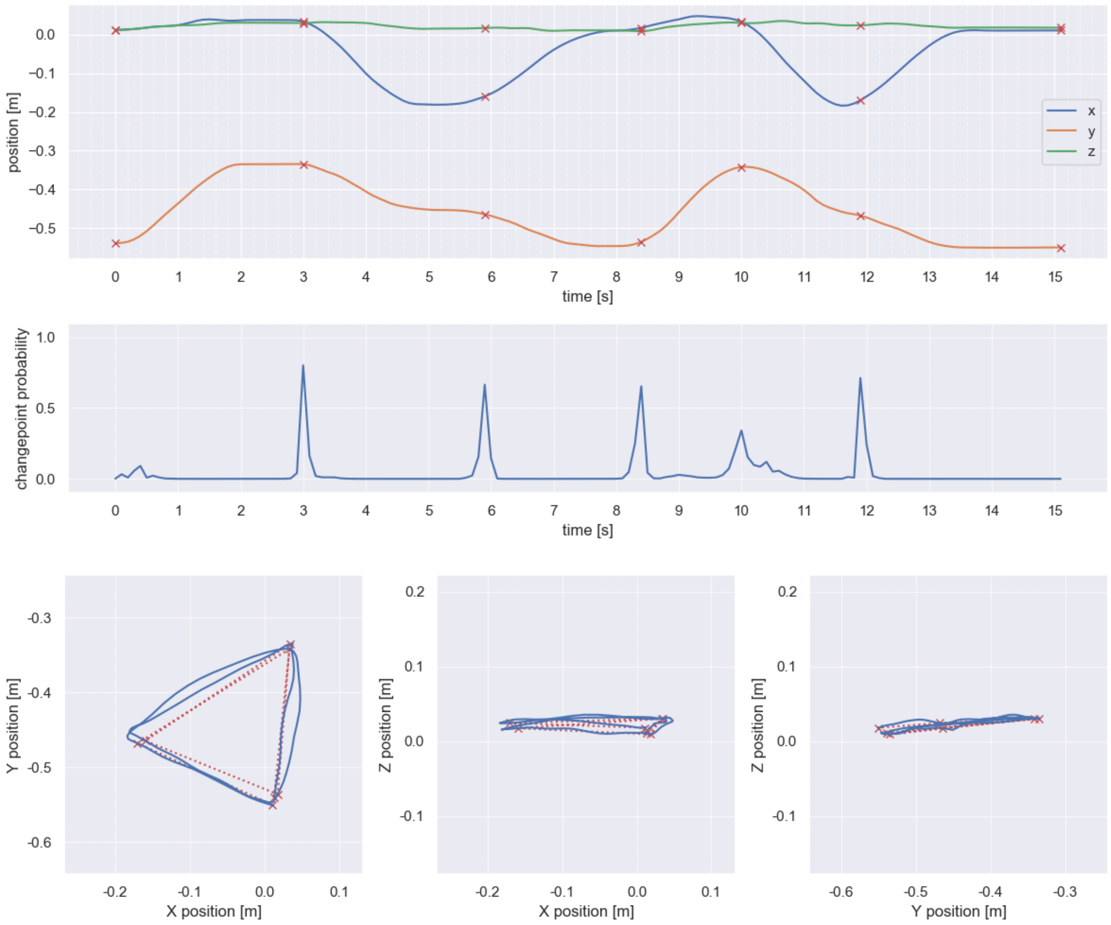
become an additional task goal. In this execution the robot end effector collide with the objects and the guider has to extend the task execution manually with a further changepoints.
Therefore, we refine the configuration parameter to the model parameter set two and rerun the teaching process. An example segmentation result of the parkour task with the second configuration set is shown in the Fig. 4.11. The new segmentation result contains additional changepoints which allow for collision-free execution.

Pick and place

An example segmentation result of the pick and place task with the first configuration set is shown in the Fig. 4.12.
By inspecting, the segments can be classified into the sub-tasks, where the segments $s_{0:1}$ can be assigned to the grasping motion with the grasping position at the desired position of the motion, the transport is represented by the segments $s_{1:2}$ and $s_{2:3}$ with the place position at the desired position of the segment $s_{2:3}$ and the end position is reached with the segment $s_{3:4}$. The segmentation result thus indicates the task structure. During several executions, the guider report issues with the pick and place position as the robot collide with the table or the object and the robot reports an collision error and stop the execution. To avoid this, the guider replace the Cartesian Motion App with the Move To Contact App, which is extended by an contact observer and therefore, the robot expect a contact and stops after detecting it.

Conclusion

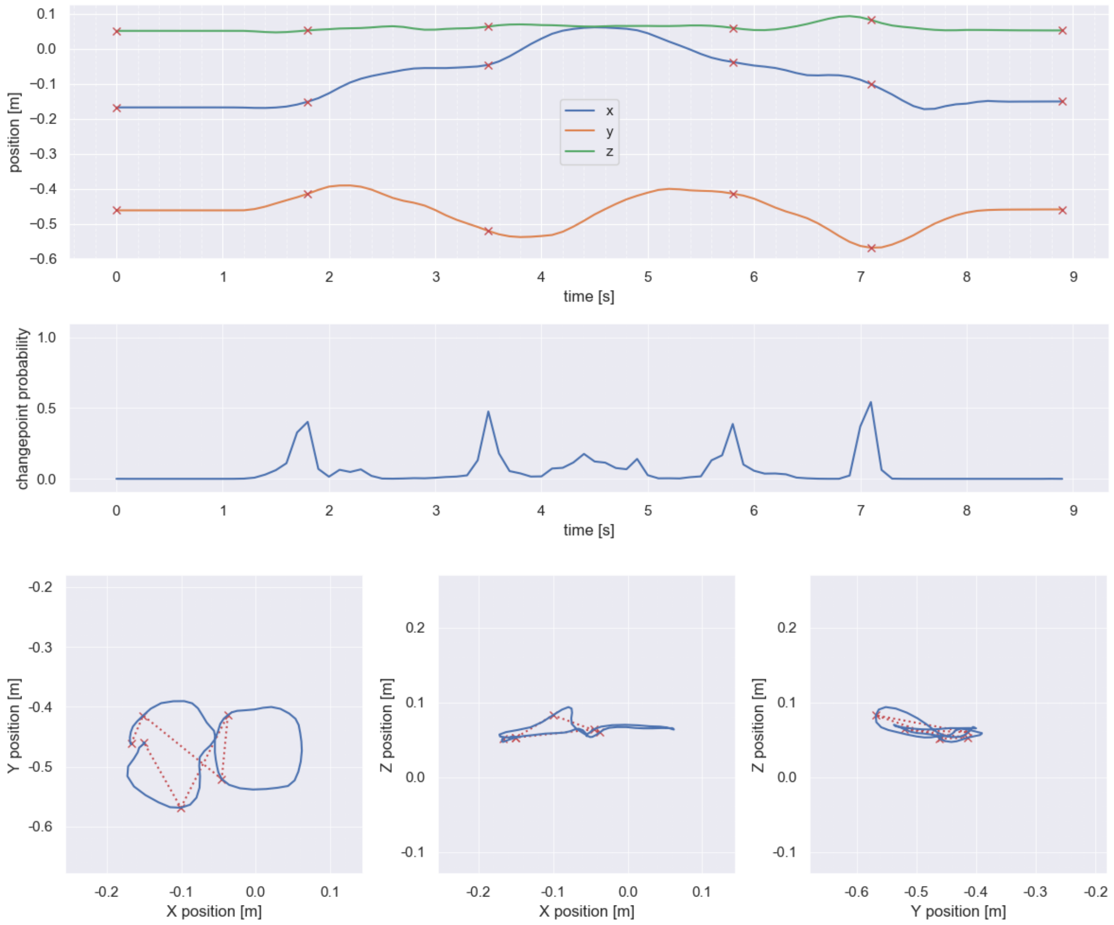
Due to the higher information level used in the definition of the Second Order Model, the segmentation report 4.9 shows the recognition of a typical movement of the robot as described in Three-Point Scenario 4.2.2 by segment $s_{0:1}$. With the additional goal of not touching the obstacle in scenario 4.2.2, the evaluation of the feedback showed that the second order model precision depends on the configuration parameters used. Although in scenario 4.2.2 the robot in free space assumption is not fulfilled during pick up, transport and place phases, the segmentation report shows a complete segmentation. Generally, the second order model enables a segmentation in the intended way and with the focus on the desired position parameter, the intention recognition convinced.



Segments Parameters

	t_0	t_n	x_0 [m]			x_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	3.0	0.013	-0.54	0.012	0.035	-0.335	0.03	SOM
$s_{1:2}$	3.0	5.9	0.035	-0.335	0.03	-0.16	-0.464	0.018	SOM
$s_{2:3}$	5.9	8.4	-0.16	-0.464	0.018	0.018	-0.536	0.01	SOM
$s_{3:4}$	8.4	10.0	0.018	-0.536	0.01	0.034	-0.342	0.031	SOM
$s_{4:5}$	10.0	11.9	0.034	-0.342	0.031	-0.17	-0.468	0.025	SOM
$s_{5:6}$	11.9	15.1	-0.17	-0.468	0.025	0.011	-0.55	0.018	SOM

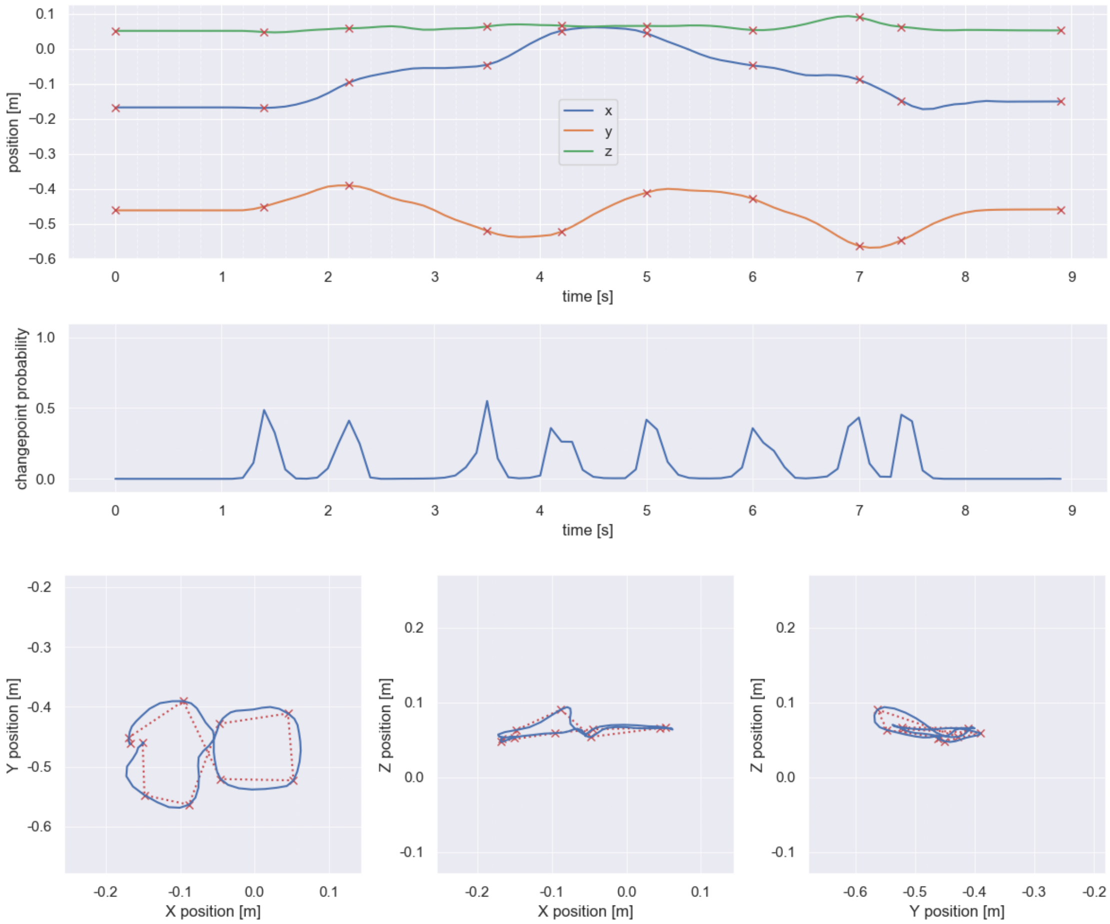
Figure 4.9: **Segmentation Report(Second Order Model with first configuration parameter set, Three-Point motion):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The third subplot shows the motion in position space. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.



Segments Parameters

	t_0	t_n	x_0 [m]			x_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	1.8	-0.167	-0.462	0.052	-0.151	-0.414	0.053	SOM
$s_{1:2}$	1.8	3.5	-0.151	-0.414	0.053	-0.046	-0.521	0.064	SOM
$s_{2:3}$	3.5	5.8	-0.046	-0.521	0.064	-0.037	-0.414	0.06	SOM
$s_{3:4}$	5.8	7.1	-0.037	-0.414	0.06	-0.1	-0.569	0.083	SOM
$s_{4:5}$	7.1	8.9	-0.1	-0.569	0.083	-0.15	-0.459	0.053	SOM

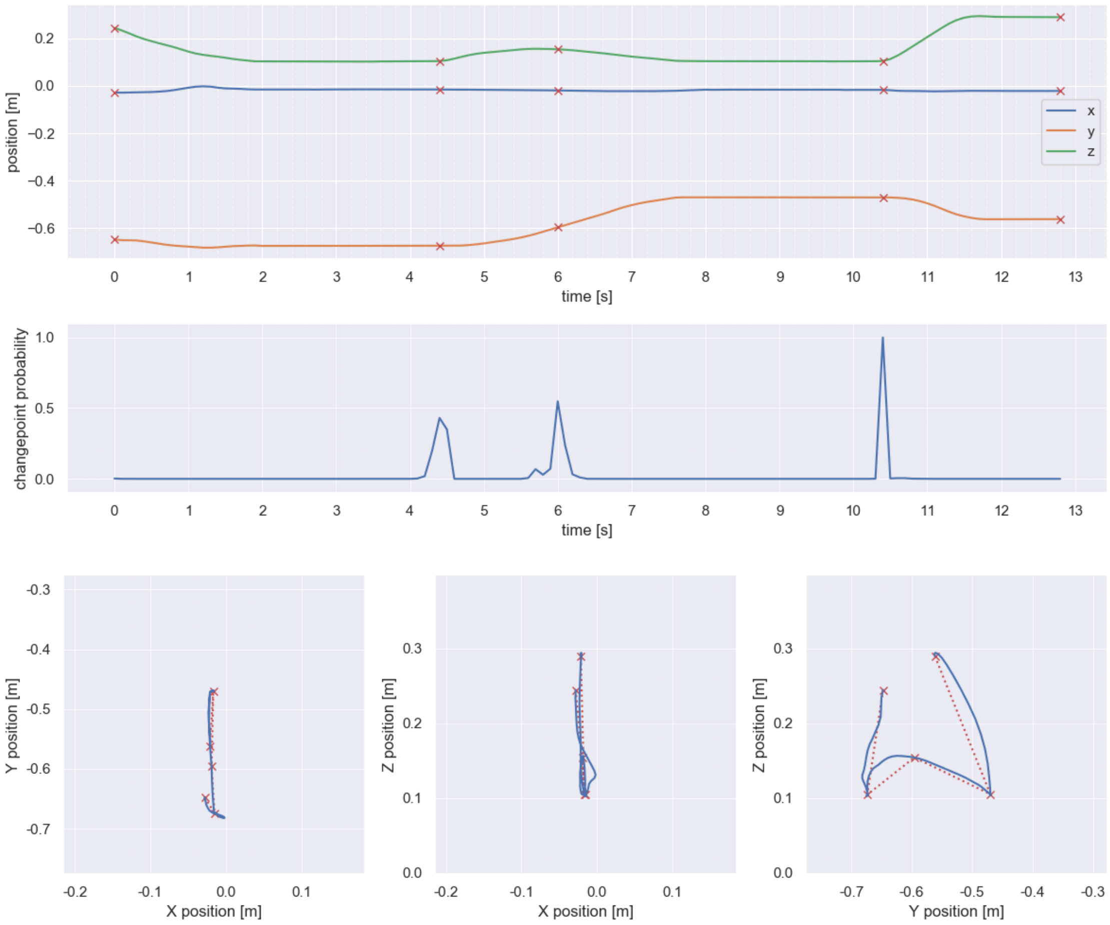
Figure 4.10: **Segmentation Report(Second Order Model with first configuration parameter set, Parkour):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The third subplot shows the motion in position space. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.



Segments Parameters

	t_0	t_n	x_0 [m]			x_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	1.4	-0.167	-0.462	0.052	-0.169	-0.451	0.048	SOM
$s_{1:2}$	1.4	2.2	-0.169	-0.451	0.048	-0.096	-0.39	0.059	SOM
$s_{2:3}$	2.2	3.5	-0.096	-0.39	0.059	-0.046	-0.521	0.064	SOM
$s_{3:4}$	3.5	4.2	-0.046	-0.521	0.064	0.052	-0.523	0.067	SOM
$s_{4:5}$	4.2	5.0	0.052	-0.523	0.067	0.045	-0.411	0.066	SOM
$s_{5:6}$	5.0	6.0	0.045	-0.411	0.066	-0.048	-0.428	0.054	SOM
$s_{6:7}$	6.0	7.0	-0.048	-0.428	0.054	-0.088	-0.563	0.091	SOM
$s_{7:8}$	7.0	7.4	-0.088	-0.563	0.091	-0.148	-0.547	0.063	SOM
$s_{8:9}$	7.4	8.9	-0.148	-0.547	0.063	-0.15	-0.459	0.053	SOM

Figure 4.11: **Segmentation Report(Second Order Model with second configuration parameter set, Parkour):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The third subplot shows the motion in position space. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.



Segments Parameters

	t_0	t_n	x_0 [m]			x_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	4.4	-0.028	-0.648	0.244	-0.015	-0.674	0.104	SOM
$s_{1:2}$	4.4	6.0	-0.015	-0.674	0.104	-0.019	-0.596	0.154	SOM
$s_{2:3}$	6.0	10.4	-0.019	-0.596	0.154	-0.017	-0.47	0.105	SOM
$s_{3:4}$	10.4	12.8	-0.017	-0.47	0.105	-0.021	-0.562	0.29	SOM

Figure 4.12: **Segmentation Report(Second Order Model with first configuration parameter set, Pick and Place):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The third subplot shows the motion in position space. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.

4.2.3 Force Model

The segmentation model is generated by presented formulation of the Force model from Sec. 3.2.2 substituted into the formulation of the BCPD algorithm 2.1 and with the parameters shown in Tab. 4.3. Because the force model ignores movements, a subsequent execution of the segmentation result is incomplete. Therefore we will focus the evaluation on the recognition of contact states and the applied forces.

Configuration Parameter Set					
N.	f_S[hz]	$c_{\text{threshold}}$	$(v_0,\,\sigma_0^2)$	K with $(r = e^K)$	f_0
1	10	0.2	$(10, 1.25^2)$	$[0, 0.15, \ldots, 3]$	$\begin{bmatrix}0. & 0. & 0.\end{bmatrix}$

Table 4.3: Configuration Parameter Set

The sample frequency f_S, for extraction threshold of the changepoints $c_{\text{threshold}}$, the conjugate prior inverse-chi-squared hyperparameters $(v_0,\, \sigma_0^2)$, first order dynamic roots r exponential distributed like K with $(r = e^K)$ and the f_0 as the fixed force start value.

Dispenser

The segmentation report for the scenario dispenser presented in Sec.4.1.1 is shown in Fig. 4.13. The segmentation result shows that the force model recognizes both the contact and the individual phases of the contact. In the segments $s_{0:1}, s_{3:4}$ and $s_{6:7}$ the robot is moving and not in contact with the soap dispenser. In segments $s_{1:2}$ and $s_{4:5}$ the dispenser is pushed to the stop with an applied force of approx 20 Newton. Because the force does not fall off immediately, the decrease is shown in the segments $s_{2:3}$ and $s_{5:6}$.

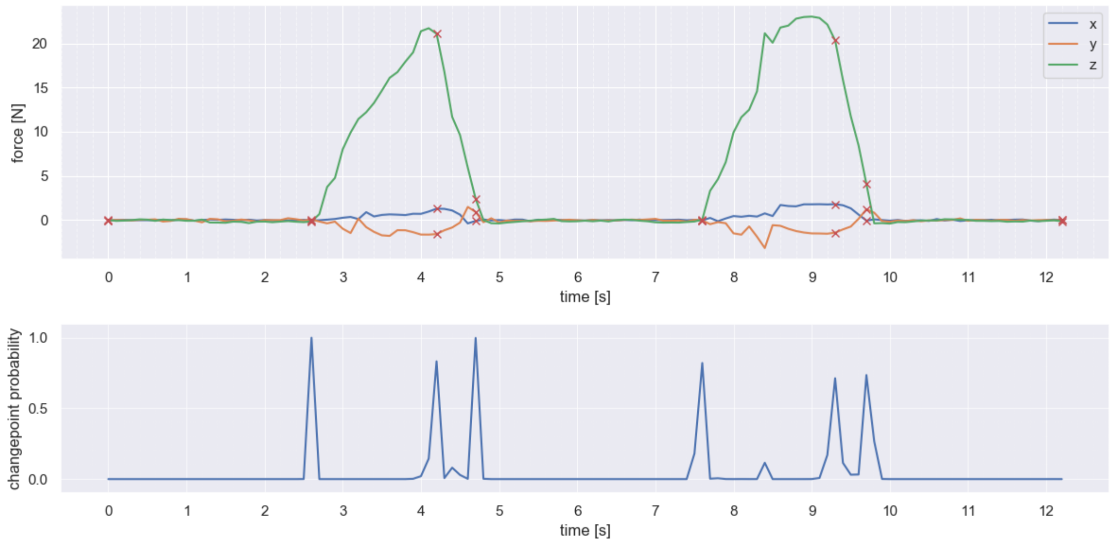
Pick and Place

The segmentation report for the scenario dispenser presented in Sec.4.1.1 is shown in Fig. 4.14. When examining the segment parameters of the segmentation result, the segments $s_{0:1}$ and $s_{6:7}$ show a low desired force, indicating that the robot is not in contact with the object during these segments. In segment $s_{1:2}$ and $s_{4:5}$ the robot is in contact with the object to pick it up and put it down. Again, the drop in force is seen in segment $s_{2:3}$ and $s_{5:6}$. In segment $s_{3:4}$ the target object is transported.

Conclusion

In summary, the force model shows the ability to recognize contact and the applied force. In the Dispenser scenario, the force vector required for successful actuation could be identified. However, based on the rigid robot and object assumption, the decrease in force is assumed to be instantaneous, which in reality results in further segments.

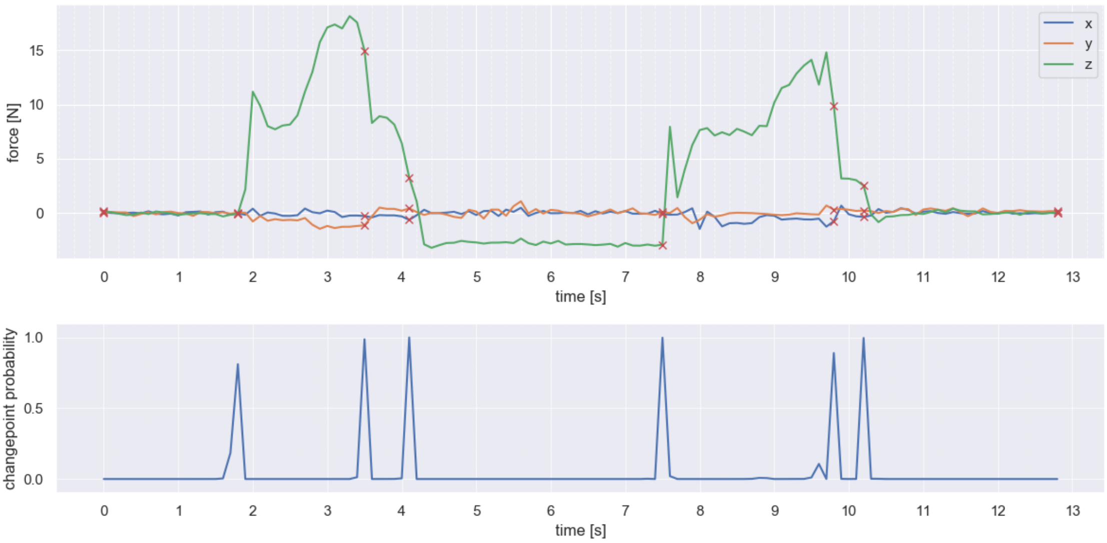
In the Pick and Place scenario the focus is on identifying the contact state. By setting a threshold value of 1 Newton the segments can be easily distinguished into those where the robot and the object are in contact or where the robot is in free space. Also compensating the weight of the target object during transport could be detected as applying a negative force.



Segments Parameters

	t_0	t_n	f_0 [m]			f_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	2.6	0	0	0	-0.032	0.062	-0.157	ForceModel
$s_{1:2}$	2.6	4.2	0	0	0	1.291	-1.581	21.086	ForceModel
$s_{2:3}$	4.2	4.7	0	0	0	-0.083	0.918	2.411	ForceModel
$s_{3:4}$	4.7	7.6	0	0	0	-0.03	-0.033	0.06	ForceModel
$s_{4:5}$	7.6	9.3	0	0	0	1.806	-1.418	20.331	ForceModel
$s_{5:6}$	9.3	9.7	0	0	0	-0.058	1.22	4.083	ForceModel
$s_{6:7}$	9.7	12.2	0	0	0	0.013	0.007	-0.134	ForceModel

Figure 4.13: **Segmentation Report(Force Model, Dispenser):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.



Segments Parameters

	t_0	t_n	f_0 [m]			f_d [m]			type
			x	y	z	x	y	z	
$s_{0:1}$	0.0	1.8	0	0	0	-0.027	0.035	-0.064	ForceModel
$s_{1:2}$	1.8	3.5	0	0	0	-0.278	-1.139	14.986	ForceModel
$s_{2:3}$	3.5	4.1	0	0	0	-0.631	0.419	3.283	ForceModel
$s_{3:4}$	4.1	7.5	0	0	0	-0.068	0.095	-2.911	ForceModel
$s_{4:5}$	7.5	9.8	0	0	0	-0.757	0.318	9.881	ForceModel
$s_{5:6}$	9.8	10.2	0	0	0	-0.358	0.231	2.524	ForceModel
$s_{6:7}$	10.2	12.8	0	0	0	-0.005	0.155	0.055	ForceModel

Figure 4.14: **Segmentation Report(Force Model, Pick and Place):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.

4.2.4 Contact Model

The segmentation model is generated by presented formulation of the contact model 3.2.3 substituted into the formulation of the BCPD algorithm 2.1 and with the parameters shown in Tab. 4.1. Due to the focus on the contact state and the lack of segmentation in movements and applied forces, the segmentation model quality is evaluated based on the accuracy of the contact state detection.

Configuration Parameter Sets							
N.	f_S[hz]	$c_{\text{threshold}}$	σ^2_{sensor}	$\sigma^2_{\text{contact}}$	$\sigma^2_{\text{transport}}$	$m_{\text{min}}[Kg]$	α_0
1	10	0.2	0.75	20	0	0.2	$\begin{bmatrix}10 & 10 & 10\end{bmatrix}$

Table 4.4: Configuration Parameter Sets

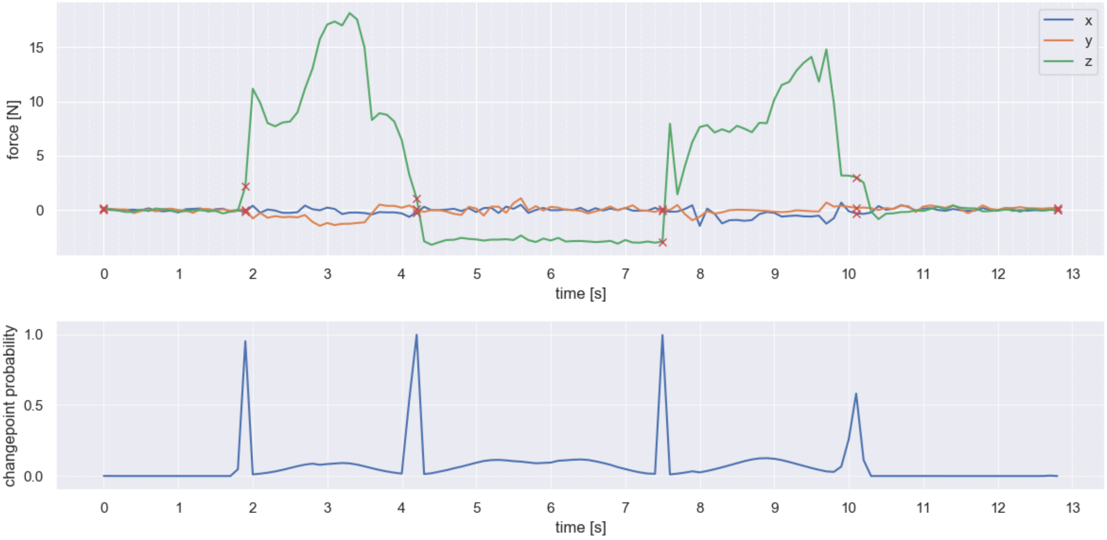
The sample frequency f_S, for extraction threshold of the changepoints $c_{\text{threshold}}$, the sensor variance σ^2_{sensor}, the variance during a contact $\sigma^2_{\text{contact}}$, the variance during the transport $\sigma^2_{\text{transport}}$ and the Dirichlet distributed conjugate prior hyperparameter vector α_0.

Pick and Place

The segmentation report for the pick and place scenario presented in Sec.4.1.1 is shown in Fig. 4.15. The segmentation result of the contact model for the pick and place task shows 5 segments in 3 different contact states. The approach to the object and the move to the final position are classified as free space in the segments $s_{0:1}$ and $s_{4:5}$. In the segments $s_{1:2}$ and $s_{3:4}$ the robot was classified as in contact with the object and a transport of an object with a weight of 276 grams was detected in the segment $s_{2:3}$.

Conclusion

With the focus on the recognition of the contact state in the defined three states Free Space, In Contact and Transport the Contact Model shows the desired results. In a first evaluation of a pick and place scenario, the Contact Model was able to recognize the task structure based on the environmental force alone.



Segments Parameters

	t_0	t_n	u	m	f_0 [m]			f_d [m]			type
					x	y	z	x	y	z	
$s_{0:1}$	0.0	1.9	1	-	0.073	-0.013	0.196	-0.158	0.036	2.182	ContactModel
$s_{1:2}$	1.9	4.2	2	-	-0.158	0.036	2.182	-0.203	0.094	1.076	ContactModel
$s_{2:3}$	4.2	7.5	3	0.276	-0.203	0.094	1.076	-0.068	0.095	-2.911	ContactModel
$s_{3:4}$	7.5	10.1	2	-	-0.068	0.095	-2.911	-0.305	0.196	3.033	ContactModel
$s_{4:5}$	10.1	12.8	1	-	-0.305	0.196	3.033	-0.005	0.155	0.055	ContactModel

Figure 4.15: **Segmentation Report(Contact Model, Pick and Place):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.

4.2.5 Fully Integrated Model

The segmentation model is generated by presented formulation of the Fully Integrated Model 3.2.4 substituted into the formulation of the BCPD algorithm. 2.1 and with the parameters shown in Tab. 4.5.
The segments of the Fully Integrated Model segmentation result are classified based on the model used and are provided with contact variables as segment parameters. The robot execution was generated based on the segment parameters, whereby movements were realized with the Cartesian motion app and applied forces with the apply force app. Since the Cartesian Motion app was designed for collision free movements, we used the contact variable transition from free motion to contact to replace the Cartesian Motion app with the Move To Contact App. If during a contact a grip or release is detected, the applied force app is replaced by the Grasp or Gripper move app.

Configuration Parameter Set								
N.	f_S [hz]	$c_{\text{threshold}}$	θ_{motion} (v_0, σ_0^2)	θ_{steady} (v_0, σ_0^2)	$\theta_{\text{no contact}}$ (v_0, σ_0^2)	θ_{contact} (v_0, σ_0^2)	$\theta_{\text{transport}}$ (v_0, σ_0^2)	m_{min} [Kg]
1	10	0.2	$(1, 0.025^2)$	$(1, 0.025^2)$	$(1, 0.5^2)$	$(1, 1.25^2)$	$(1, 0.5^2)$	0.2

Table 4.5: Configuration Parameter Set of the Combined Model
The sample frequency f_S, for extraction threshold of the changepoints $c_{\text{threshold}}$. The θ_{motion}, θ_{steady}, $\theta_{\text{no contact}}$, θ_{contact} and $\theta_{\text{transport}}$ are the individual contact model parameter sets. The m_{min} is the minimum weight.

Apply Dispenser

The segmentation report for the Dispenser scenario presented in Sec.4.1.1 is shown in Fig. 4.16. The segmentation result of the FIM describes the dispenser scenario in the form of classified and parameterized segments, which can be interpreted on the basis of the task description. Thus, the movements in the segments $s_{0:1}$, $s_{3:4}$, $s_{4:5}$ and $s_{7:8}$ can be described as the approach to the dispenser, the set-down and renewed approach to the dispenser and the movement to the final position. In the segments $s_{1:2}$ and $s_{5:6}$ the robot is classified as in contact and applies a force of about 20 Newton, which describes the pressing of the dispenser. The segments $s_{2:3}$ and $s_{6:7}$ can be explained by the decrease in force and the assumptions of the rigid objects and robot in the definition of the force model.
For the execution the segments $s_{0:1}$ and $s_{4:5}$ are realized based on the Contact variable transition with the Move To Contact app. The pressing of the dispenser in the segments $s_{1:2}$ and $s_{5:6}$ is realized with the Apply Force App and the desired force as applied force vector. With a minimum time threshold filter the segments $s_{2:3}$ and $s_{6:7}$ were filtered out and excluded in the execution. The movement in the segment $s_{3:4}$ was realized with the Cartesian Motion App and the desired position. To prevent damage to the robot and the soap dispenser the power is limited to 10

Newton. For this purpose the force Vector is scaled appropriately when the Applied Force App is parameterized.

Despite the force limitation, test users describe the execution generated on the basis of the segmentation result of the FIM as a successful pressing of the dispensing. Since the manual generation of an execution as in the dispenser scenario requires the user to estimate the force, experienced users describe the recognition of the force vector as a special added value.

Pick and Place:

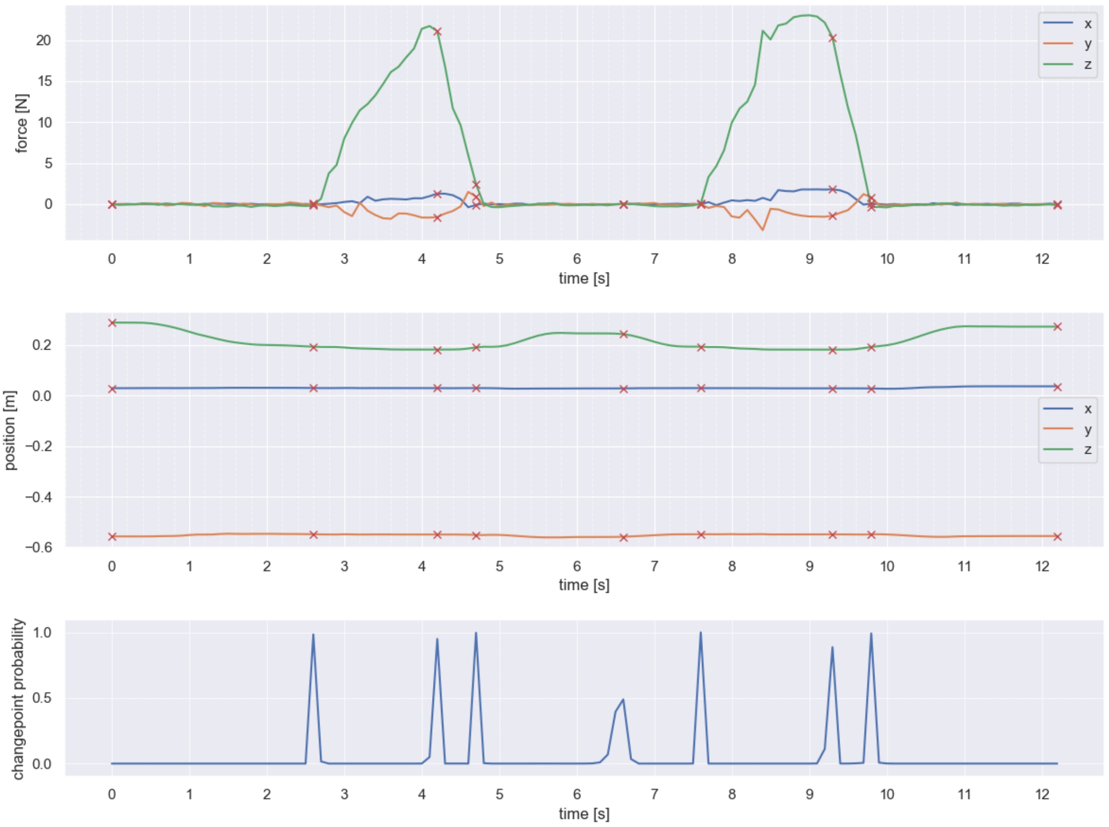
The segmentation report for the pick and place with a payload of 250 grams scenario presented in Sec.4.1.1 is shown in Fig. 4.17. By comparing the segments of the segmentation result of the Fully Integrated Model with the task structure of the pick and place tasks, the individual sub-tasks show up as classified parameterized segments. The approach to the target object and the motion to the finale position is classified as a second order dynamic motion with the desired positions in the segments $s_{0:1}$ and $s_{8:9}$. The contact during the object grab and release is classified as contact in the segments $s_{1:2}$ and $s_{5:6}$. The transport was recognized as a sequence of two movements in the segments $s_{3:4}$ and $s_{4:5}$ and parameterized with an object weight of 257 grams and 293 grams. In the segments $s_{2:3}$ and $s_{6:7}$ the decreasing force leads to segments that are classified as movements with a target position similar to the start position.

For the execution the motion in segment $s_{0:1}$ is realized by a Move to Contact app. With the grasping detected at time 3s and the opening of the gripper detected at time 9.2s, the Grasp and Gripper Move app replaced the segments during the contacts. The transport segments are realized with the help of the transport app and the average weight of 273 grams.

During the execution of the test users the pick and place scenario could be executed successfully each time.

Conclusion

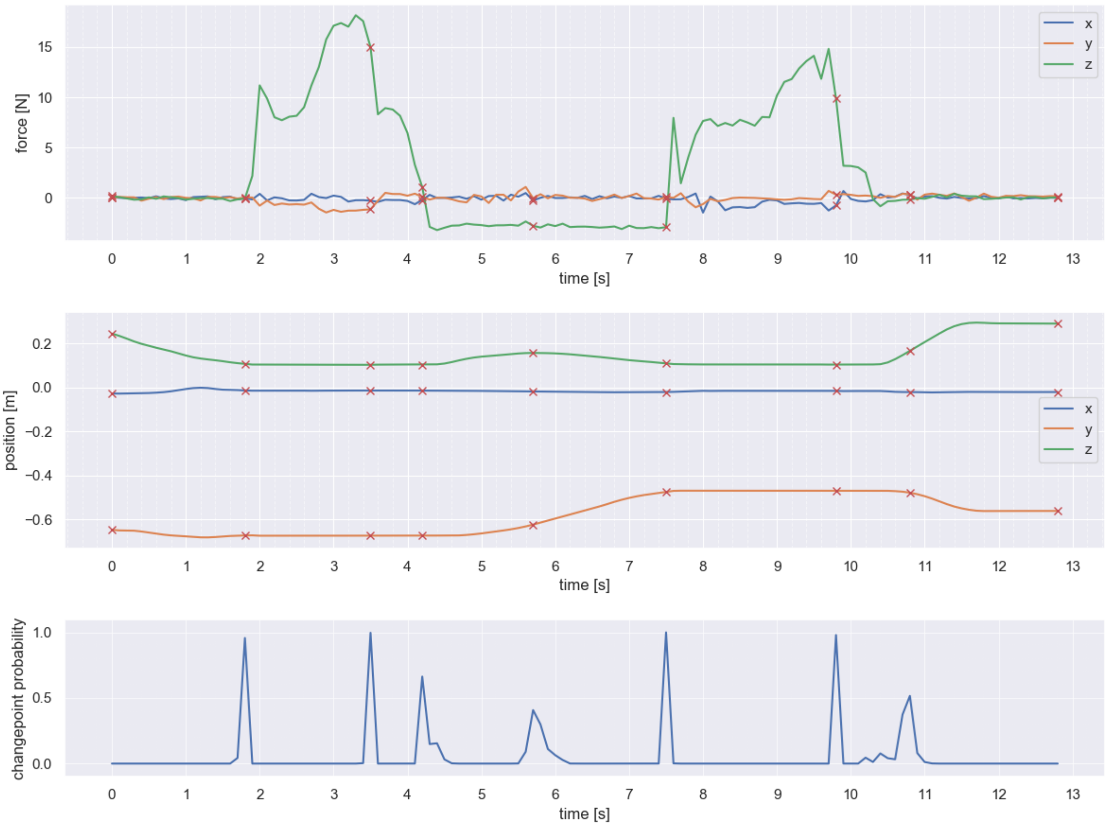
As a combination of the force model, the second order dynamic model and the contact model, the Fully Integrated Model demonstrates the segment result of all models and is therefore able to detect movements, applied forces and contact states. In addition, the Fully Integrated Model is based on the position space and force space and can therefore interpret information from both signals in combination. The recognition of the structure of the task based on a sequence of classified and parameterized segments could be proven in the evaluation of introduced scenarios. In addition, a detailed robot task description and execution has been generated, which is based on an interpretation of the segment parameters.



Segments Parameters

	t_0	t_n	u	m	x_0 [m]			x_d [m]			type	f_0 [m]			f_d [m]			type
					x	y	z	x	y	z		x	y	z	x	y	z	
$s_{0:1}$	0.0	2.6	1	-	0.028	-0.558	0.288	0.029	-0.549	0.192	SOM	-0.047	-0.038	0.029	-0.032	0.062	-0.157	No Contact
$s_{1:2}$	2.6	4.2	2	-	0.029	-0.549	0.192	0.028	-0.55	0.181	Zero	0.0	0.0	0.0	1.291	-1.581	21.086	FM
$s_{2:3}$	4.2	4.7	2	-	0.028	-0.55	0.181	0.029	-0.552	0.19	Zero	0.0	0.0	0.0	-0.083	0.918	2.411	FM
$s_{3:4}$	4.7	6.6	1	-	0.029	-0.552	0.19	0.027	-0.559	0.242	SOM	-0.083	0.918	2.411	0.021	0.024	0.055	No Contact
$s_{4:5}$	6.6	7.6	1	-	0.027	-0.559	0.242	0.029	-0.549	0.192	SOM	0.021	0.024	0.055	-0.03	-0.033	0.06	No Contact
$s_{5:6}$	7.6	9.3	2	-	0.029	-0.549	0.192	0.028	-0.55	0.181	Zero	0.0	0.0	0.0	1.806	-1.418	20.331	FM
$s_{6:7}$	9.3	9.8	2	-	0.028	-0.55	0.181	0.027	-0.55	0.191	Zero	0.0	0.0	0.0	0.086	0.834	-0.368	FM
$s_{7:8}$	9.8	12.2	1	-	0.027	-0.55	0.191	0.036	-0.556	0.272	SOM	0.086	0.834	-0.368	0.013	0.007	-0.134	No Contact

Figure 4.16: **Segmentation Report(Fully Integrated Model, Dispenser):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.



Segments Parameters

	t_0	t_n	u	m	x_0 [m]			x_d [m]			type	f_0 [m]			f_d [m]			type
					x	y	z	x	y	z		x	y	z	x	y	z	
$s_{0:1}$	0.0	1.8	1	-	-0.028	-0.648	0.244	-0.014	-0.673	0.107	SOM	0.073	-0.013	0.196	-0.027	0.035	-0.064	No Contact
$s_{1:2}$	1.8	3.5	2	-	-0.014	-0.673	0.107	-0.015	-0.674	0.102	Zero	0.0	0.0	0.0	-0.278	-1.139	14.986	FM
$s_{2:3}$	3.5	4.2	1	-	-0.015	-0.674	0.102	-0.015	-0.674	0.104	SOM	-0.278	-1.139	14.986	-0.203	0.094	1.076	No Contact
$s_{3:4}$	4.2	5.7	3	0.257	-0.015	-0.674	0.104	-0.018	-0.624	0.156	SOM	-0.203	0.094	1.076	-0.267	-0.037	-2.777	Transport
$s_{4:5}$	5.7	7.5	3	0.293	-0.018	-0.624	0.156	-0.021	-0.475	0.109	SOM	-0.267	-0.037	-2.777	-0.068	0.095	-2.911	Transport
$s_{5:6}$	7.5	9.8	2	-	-0.021	-0.475	0.109	-0.016	-0.47	0.104	Zero	0.0	0.0	0.0	-0.757	0.318	9.881	FM
$s_{6:7}$	9.8	10.8	1	-	-0.016	-0.47	0.104	-0.021	-0.479	0.166	SOM	-0.757	0.318	9.881	0.293	0.356	-0.168	No Contact
$s_{7:8}$	10.8	12.8	1	-	-0.021	-0.479	0.166	-0.021	-0.562	0.29	SOM	0.293	0.356	-0.168	-0.005	0.155	0.055	No Contact

Figure 4.17: **Segmentation Report(Fully Integrated Model, Pick and Place):** This figure illustrates the segmentation report consisting of the acquired robot signals in the first subplot followed by the changepoint probability, which is the result of the segmentation with the Bayesian changepoint detection algorithms. The evaluated changepoints are marked as red crosses and connected with a red dotted line. The table lists the segment parameters for the selected model and the segmentation result.

4.2.6 Online Segmentation

With the formulation of the oBCPD algorithm presented in Sect. 2.1.5 we will investigate an online segmentation based on the combined model. For this purpose we will compare the segmentation results of the offline segmentation and the online segmentation. The common representation of the segmentation results of the oBCPD algorithm are the growth probabilities of increasing run lengths. With each new observation the run length probabilities are updated and a changepoint is shown by shifting the probabilities from high run lengths to low ones. With further observations the segment parameters become more precise and the position of a changepoint stabilizes with a delay. Therefore, we will consider a delay in comparing segmentation results and online segmentation results.

Contact Model

Figure 4.18 shows the segmentation results of the offline segmentation and the online segmentation of the pick and place scenario based on the contact model.
With a delayed evaluation after 0.6 seconds, all four change points of the offline segmentation are detected in the online segmentation. The calculation time of a cycle with a non-optimized implementation averages 18,047 milliseconds and is therefore far below the sampling rate of 100 milliseconds. Thus, the online segmentation result based on the contact model with the against parameters and with an evaluation delay of 0.6 seconds already shows a segmentation result which corresponds to the offline segmentation by evaluating the run length probability.

Fully Integrated Model

Figure 4.19 shows the segmentation results of the offline segmentation and the online segmentation of the pick and place scenario based on the Fully Integrated Model.
The offline segmentation result recognized changepoints, which are based on a contact state change, i.e. the changepoints after 1.8, 4.2, 7.4 and 10.2 seconds are also recognized in the online segmentation result with evaluation delays of 0.6 after 2, 4.2, 7.8 and 10.6 seconds. The changepoints detected during the contact after 3.5 and 9.8 seconds in the offline segmentation result could not be detected in the online segmentation result. The changepoint generated in the transporting phase after 6 seconds could only be detected with the second configuration parameter set for the second order motion model in Tab. 4.2 and after an evaluation delay of 1.2 seconds. The calculation time of a cycle with a non-optimized implementation averages 82.6 milliseconds and is therefore also below the sampling rate of 100 milliseconds.

Conclusion

Within the evaluation of the online segmentation for the Contact Model and the Fully Integrated Model it could be shown for the pick and place scenario that with an

evaluation delay of 0.6 seconds changes in the contact state can already be detected during the execution. The Fully Integrated Model, which is based on the Force Model and the Second Order Model, requires an adjustment of the model parameters with respect to changepoints within a contact state. In addition, these changepoints require an extended evaluation delay time of 1.2 seconds. Due to the evaluation delay of 1.2 seconds, changepoints of small segments cannot be detected, which is evident for the changepoints after 3.4 and 9.8 seconds.
With a cycle calculation time below the sampling rate it could be shown that with online segmentation the detection of changepoints during execution is possible.

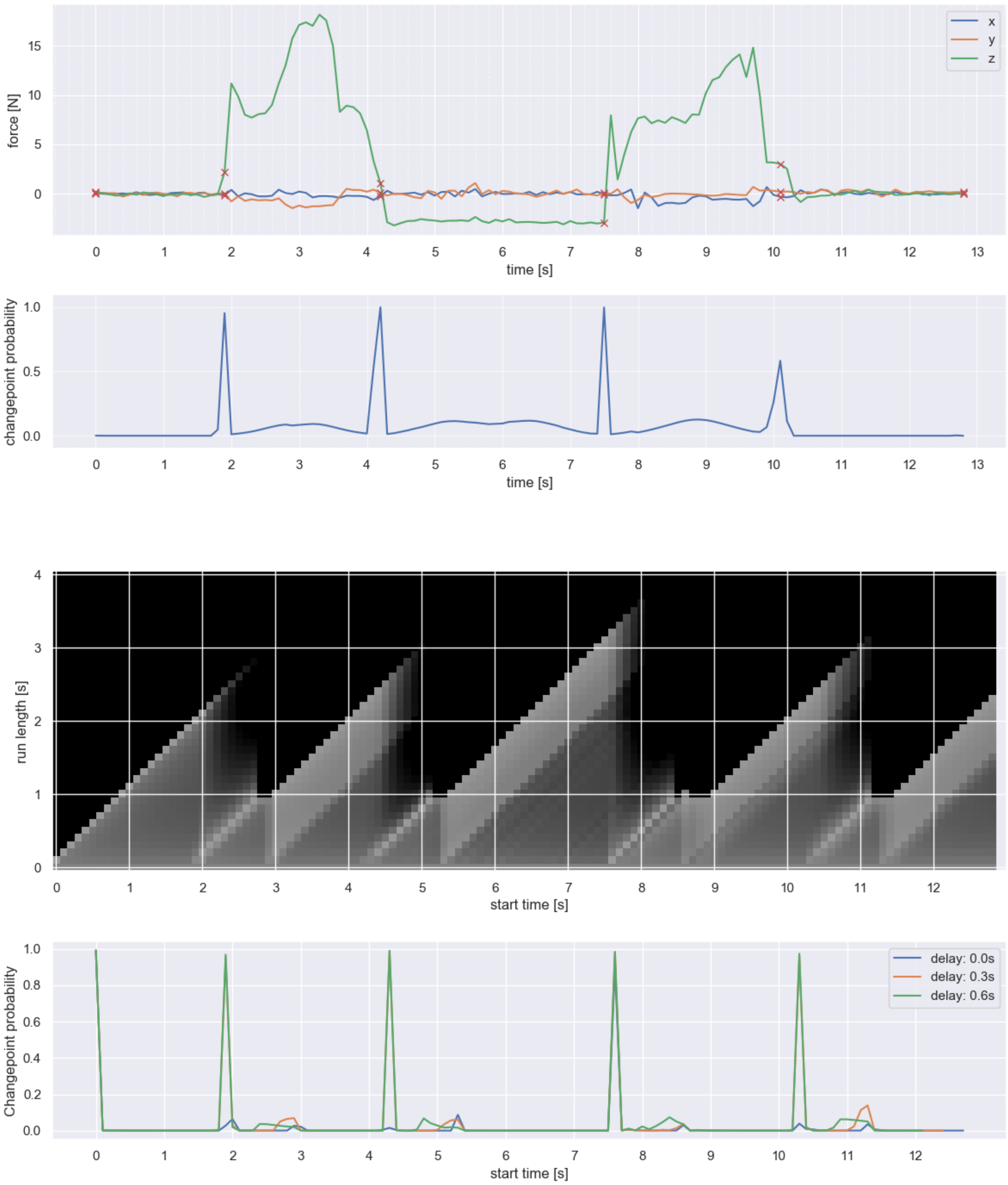


Figure 4.18: The two three subplots show the acquired position signal of the pick and place scenario followed by the offline segmentation results. The third subplot illustrates the increasing run length probability for each start time as a grey value, with higher probabilities being displayed brighter. The fourth subplot illustrates different run length probabilities shifted to the segment start.

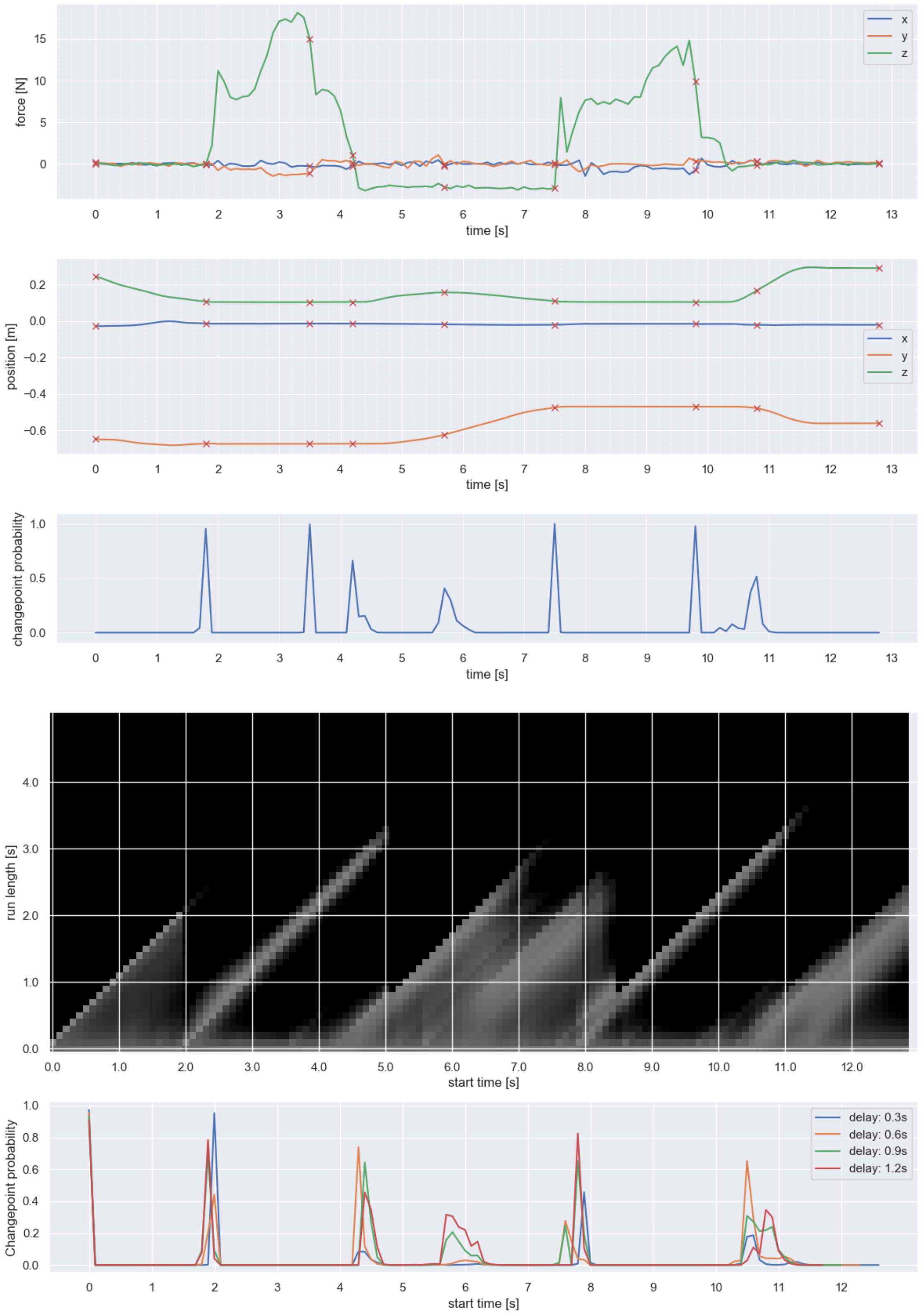


Figure 4.19: The top three subplots show the acquired position and force signal of the pick and place scenario followed by the offline segmentation result. The fourth subplot illustrates the increasing run length probability for each start time as a grey value, with higher probabilities being displayed brighter. The fifth subplot illustrates different run length probabilities shifted to the segment start.

4.3 Discussion

In summary, it could be shown that a segmentation based on segment classification for a limited number of scenarios recognizes the task structure and the intention goals. Different Informed Models have been generated based on assumptions that simulate the real Robot Dynamic, environment and human demonstrator. However, the used configuration parameters show an influence on the segmentation precision, which means that the parameters have to be adjusted according to the task requirements. Furthermore, the use case was tailored to a 7- Degree of Freedom (DOF) robot arm with a gripper and a guideline was generated for the Kinaesthetic robot teaching, which defines contact points with objects and humans.

Chapter 5

Conclusion

In the experiments we demonstrated that the introduced segmentation approach as a combination of simultaneous segment identification and classification based on the Informed Segment Model and the BCPD algorithm provides results, which have achieved a good feedback in the context of a first intention recognition interpretation and execution. With the presented model formulation of the second and first order dynamic, the focus could be set to the desired position or the desired force as model parameter, which optimizes the segmentation result regarding the assumption of the intention goals. However, it is demonstrated that the segmentation accuracy is dependent on the selected model parameters and with the lack of sensitivity to the ratio of the selected velocity and the desired precision, especially for tasks with a varying error tolerance, no model parameter can be found that can both detect precise movements and filter out non-intended movements. As demonstrated in the soap dispenser experiment, the presented force model recognizes the applied force and despite the assumed restriction of no environmental information, the task structure can still be classified. Since a task consists of a sequence of sub-tasks which often differ in the contact state, a basic task structure is recognized by its identification. Such a task structure identification of the different contact states could be realized in the context of the experiment for the following contact states: free space, in contact and transport and for a pick an place task and the soap dispenser scenario. The segmentation result of the Second Order Model and the Force Model, and thus also the segmentation based on the contact state change from the contact model, are all shown in the segmentation result of the Fully Integrated Model. To evaluate the online segmentation, the segmentation result is compared with the offline segmentation result. For the pick and place scenario and the contact model, all changepoints are also detected in the online segmentation result with an evaluation delay of 0.6 seconds. The decisive changepoints from the offline segmentation result are reflected in the online segmentation result by adjusting the model parameters of the Fully Integrated Model and with an evaluation delay of 1.2 seconds.

In the context of a subsequent work, a direct comparison with other segmentation techniques would be the next emulation step. In order to prove the segmentation result regarding the generic validity, further setups and especially another robot arm as well as new scenarios would be crucial. With the help of the desired velocity parameter, human model can be extended to formulate the precision in position space as a function of velocity. An Informed Model based on the desired velocity parameter is able to deal with different task requirements regarding precision.

Appendix A

Appendix

A.1 Style and Expressions

A.1.1 Sets

$\mathbb{N}_0^+$ positive natural numbers
$\mathbb{N}^+$ positive natural numbers without 0
$\mathbb{R}^+$ positive real numbers without 0

A.1.2 Detailed Derivation of the Constant Velocity Model

The marginal likelihood

By collection of the prior Eq. (3.15) and Eq. (3.16) can be written like

$$\begin{aligned}
P(\boldsymbol{\beta},\sigma) &\propto N(\boldsymbol{\beta}|0,\sigma^2/\kappa_0)\mathcal{X}^{-2}(\sigma^2|v_0,\sigma_0^2)\\
P(\boldsymbol{\beta},\sigma) &= \frac{1}{\sqrt{(2\pi)^m|\sigma^2/\kappa_0|}}e^{-\frac{1}{2}(0-\boldsymbol{\beta})^T(\sigma^2/\kappa_0)^{-1}(0-\boldsymbol{\beta})}\\
&\times \quad \frac{(\sigma_0^2 v_0/2)^{v_0/2}}{\Gamma(v_0/2)}\sigma^{-2(1+v_0/2)}e^{(-v_0\sigma_0^2)/(2\sigma^2)}\\
&= \frac{1}{Z_{N\mathcal{X}^{-2}}(0,\kappa_0,V_0,\sigma_0)}\sigma^{-2/2}\sigma^{-2(1+v_0/2)}e^{-\frac{1}{2}(-\boldsymbol{\beta}^T(\sigma^2/\kappa_0)^{-1}-\boldsymbol{\beta})+(v_0\sigma_0^2)/(2\sigma^2))}\\
&= \frac{1}{Z_{N\mathcal{X}^{-2}}(0,\kappa_0,V_0,\sigma_0)}\sigma^{-2\frac{(3+v_0)}{2}}e^{-\frac{1}{2\sigma}(\kappa_0\boldsymbol{\beta}^T\boldsymbol{\beta}+v_0\sigma_0^2)}\\
Z_{N\mathcal{X}^{-2}}(0,\kappa_0,V_0,\sigma_0^2) &= (2\pi)^{\frac{m}{2}}\kappa_0^{-\frac{1}{2}}\Gamma\Big(\frac{v_0}{2}\Big)\Big(\frac{2}{\sigma_0^2 v_0}\Big)^{\frac{v_0}{2}}\ ,
\end{aligned} \tag{A.1}$$

where $Z_{N\mathcal{X}^{-2}}(0,\kappa_0,V_0,\sigma_0^2)$ is the normalization term.
For a better transparency, the following distribution formulation are determined separated by the normalization term, where the unnormalized probabilities are labeled

like $\tilde{P}$. The likelihood (3.14) reduced by the normalization term $Z_{\mathcal{N}} = \frac{1}{\sqrt{2\pi}}$ with the unnormalized formulation of the collected prior (A.1) can be written like

$$\begin{aligned} P(\beta, \sigma | \boldsymbol{x}_{1:n}, \boldsymbol{\Phi}) &\propto N(\boldsymbol{x}_{1:n} | \boldsymbol{\beta}\psi, I\sigma^2) N\mathcal{X}^{-2}(\boldsymbol{\beta}, \sigma^2 | 0, \kappa_0, v_0, \sigma_0^2) \\ \tilde{P}(\beta, \sigma | \boldsymbol{x}_{1:n}, \boldsymbol{\Phi}) &= e^{-\frac{1}{2}(\boldsymbol{x}_{1:n} - \boldsymbol{\beta\Phi})^T (\sigma^2/\kappa_0)^{-1} (\boldsymbol{x}_{1:n} - \boldsymbol{\beta\Phi})} \sigma^{-2\frac{(3+v_0)}{2}} e^{-\frac{1}{2\sigma}(\kappa_0 \boldsymbol{\beta}^T \boldsymbol{\beta} + v_0 \sigma_0^2)} \\ &= \sigma^{-2\frac{(4+v_0)}{2}} e^{-\frac{1}{2\sigma}(\kappa_0 \boldsymbol{\beta}^T \boldsymbol{\beta} + v_0 \sigma_0^2 + (Y - \boldsymbol{\beta} X)^T (Y - \boldsymbol{\beta} X))} \ , \end{aligned} \tag{A.2}$$

with the normalization term formulated like

$$Z_{N\mathcal{X}^{-2}}(\mu_n, \kappa_n, v_n, \sigma_n^2) = (2\pi)^{\frac{1}{2}} (2\pi)^{\frac{m}{2}} \kappa_0^{-\frac{1}{2}} \Gamma\left(\frac{v_0}{2}\right) . \tag{A.3}$$

Therefore, the posterior can be written like

$$P(\beta, \sigma | \boldsymbol{x}_{1:n}, \boldsymbol{\Phi}) = \frac{1}{Z_{N\mathcal{X}^{-2}}(\mu_n, \kappa_n, v_n, \sigma_n^2)} \sigma^{-2\frac{(4+v_0)}{2}} e^{-\frac{1}{2}(\boldsymbol{x}_{1:n} - \boldsymbol{\beta\Phi})^T (\sigma^2/\kappa_0)^{-1} (\boldsymbol{x}_{1:n} - \boldsymbol{\beta\Phi})} . \tag{A.4}$$

Based on the rigid linear regression formulation $\boldsymbol{\beta}^\star = (\boldsymbol{\Phi}^T \boldsymbol{\Phi} - \kappa_0 \boldsymbol{I})^{-1} \boldsymbol{\Phi} \boldsymbol{x}_{1:n}$, where $\boldsymbol{\beta}^\star$ is the optimal coefficient, the squared residual computation term can be formulated like

$$\begin{aligned} \sum_{i=1}^{n} (x_i - \boldsymbol{\beta}\phi_i)^2 &= \sum_{i=1}^{n} (x_i - \boldsymbol{\beta}^\star \phi_i)^2 + \sum_{i=1}^{n} (\boldsymbol{\beta}\phi_i - \boldsymbol{\beta}^\star \phi_i)^2 - 2 \sum_{i=1}^{n} (x_i - \boldsymbol{\beta}^\star \phi_i)(\boldsymbol{\beta}\phi_i - \boldsymbol{\beta}^\star \phi_i) \\ &= \sum_{i=1}^{n} (x_i - \boldsymbol{\beta}^\star \phi_i)^2 - (\boldsymbol{\beta} - \boldsymbol{\beta}^\star)^T (\boldsymbol{\beta} - \boldsymbol{\beta}^\star) \sum_{i=1}^{n} (\phi_i)^2 - 0 \\ &= (\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star \boldsymbol{\Phi})^T (\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star \boldsymbol{\Phi}) + \boldsymbol{\Phi}^T \boldsymbol{\Phi} (\boldsymbol{\beta} - \boldsymbol{\beta}^\star)^T (\boldsymbol{\beta} - \boldsymbol{\beta}^\star) \ , \end{aligned} \tag{A.5}$$

where here ϕ_i indicates the column vector $[1, \boldsymbol{t}_i]$ of the design matrix $\boldsymbol{\Phi}$ and the term $2\sum_{i=1}^{n} (x_i - \boldsymbol{\beta}^\star \phi_i)(\boldsymbol{\beta}\phi_i - \boldsymbol{\beta}^\star \phi_i) = 0$ dissolve, because of $\sum_{i=1}^{n} x_i = \sum_{i=1} \boldsymbol{\beta}^\star \phi_i$. Therefore, the exponent of the exponential function can be reformulated like

$$\begin{aligned} & v_0 \sigma_0^2 + \kappa_0 \boldsymbol{\beta}^T \boldsymbol{\beta} + (Y - \boldsymbol{\beta\Phi})^T (\boldsymbol{x}_{1:n} - \boldsymbol{\beta\Phi}) \\ & = v_0 \sigma_0^2 + \kappa_0 (\boldsymbol{\beta}^\star - \boldsymbol{\beta})^T (\boldsymbol{\beta}^\star - \boldsymbol{\beta}) + ns^2 + \boldsymbol{\Phi}^T \boldsymbol{\Phi} (\boldsymbol{\beta} - \boldsymbol{\beta}^\star)^T (\boldsymbol{\beta} - \boldsymbol{\beta}^\star) \\ & = v_0 \sigma_0^2 + \kappa_0 (\boldsymbol{\beta}^{\star^T} \boldsymbol{\beta}^\star - 2\boldsymbol{\beta}^{\star^T} \boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{\beta}) + ns^2 + \boldsymbol{\Phi}^T \boldsymbol{\Phi} (\boldsymbol{\beta}^T \boldsymbol{\beta} - 2\boldsymbol{\beta}^\star \boldsymbol{\beta} + \boldsymbol{\beta}^{\star T} \boldsymbol{\beta}^\star) \\ & = \boldsymbol{\beta}^T \boldsymbol{\beta} (\kappa_0 + \boldsymbol{\Phi}^T \boldsymbol{\Phi}) - 2\boldsymbol{\beta}((\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \kappa_0 I)\boldsymbol{\beta}^\star) + (v_0 \sigma_0^2 + \kappa_0 \boldsymbol{\beta}^{\star^T} \boldsymbol{\beta}^\star + ns^2 + \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{\beta}^{\star T} \boldsymbol{\beta}^\star) \ , \end{aligned} \tag{A.6}$$

with $ns^2 = (\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star \boldsymbol{\Phi})^T (\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star \boldsymbol{\Phi})$.

By comparing the formulation with a quadratic term of a conjugate prior written like

$$\begin{aligned} Q_n(\mu) &= s_n + \kappa_n (\mu - \mu_n)^2 \\ &= \mu^2 \kappa_n - 2\mu(\kappa_n \mu_n) + (\kappa_n \mu_n^2 + s_n) \ , \end{aligned} \tag{A.7}$$

where the mean defined like $\mu = \boldsymbol{\beta}$. Therefore, The hyperparameter update can be written like

$$\begin{aligned}
\kappa_n &= \mathbf{\Phi}^T\mathbf{\Phi} + \kappa_0 I \\
\kappa_n\mu_n &= (\mathbf{\Phi}^T\mathbf{\Phi} + \kappa_0 I)\boldsymbol{\beta}^\star \\
\mu_n &= \boldsymbol{\beta}^\star \\
s_n + \kappa_n\mu_n^2 &= v_0\sigma_0^2 + \kappa_0{\boldsymbol{\beta}^\star}^T\boldsymbol{\beta}^\star + (\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star\mathbf{\Phi})^T(\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star\mathbf{\Phi}) + \mathbf{\Phi}^T\mathbf{\Phi}\boldsymbol{\beta}^{\star T}\boldsymbol{\beta}^\star \\
s_0 &= v_0 + \sigma_0^2 \\
v_n &= v_0 + n \\
s_n &= v_0\sigma_0^2 + \kappa_0\boldsymbol{\beta}^{\star T}\boldsymbol{\beta}^\star + (\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star\mathbf{\Phi})^T(\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star\mathbf{\Phi}) \\
\sigma_n^2 &= \frac{s_n}{v_n}\ .
\end{aligned} \tag{A.8}$$

and the formulation of the posterior can now be written like

$$P(\boldsymbol{\beta}, \sigma|\boldsymbol{x}_{1:n}, \mathbf{\Phi}) = \frac{1}{\sqrt{2\pi|\kappa_n|}}\Gamma(v_n/2)\Big(\frac{2}{v_n\sigma_n^2}\Big)^{v_n/2} e^{-\frac{1}{2\sigma_n^2}[v_n\sigma_n^2 + \kappa_n(\boldsymbol{x}_{1:n} - \boldsymbol{\beta}^\star\mathbf{\Phi})^2]}\ . \tag{A.9}$$

The posterior predictive

For a new data point x_j, the posterior predictive distribution can be written like

$$
\begin{aligned}
P(x_j|\boldsymbol{x}_{1:n},\boldsymbol{\Phi},\phi_j) &= \frac{p(x_j,\boldsymbol{x}_{1:n}|\boldsymbol{\Phi})}{p(\boldsymbol{x}_{1:n}|\boldsymbol{\Phi})} \\
&= \frac{N\mathcal{X}^{-2}(x_j|x,\mu_{n+1},\kappa_{n+1},V_{n+1},\sigma^2_{n+1})}{N\mathcal{X}^{-2}(\mu_n,\kappa_n,V_n,\sigma^2_n)} \\
&= \frac{\Gamma(v_{n+1}/2)|\kappa_{n+1}|^{-1/2}(v_0\sigma_0^2/2)^{v_0/2}}{\Gamma(v_0/2)|\kappa_0|^{-m/2}(v_n\sigma^2_n+\frac{\kappa_n}{\kappa_{n+1}}(x_j-\boldsymbol{\beta}^\star x)^2)^{v_{n+1}/2}}\frac{1}{(\pi)^{n+1/2}} \\
&\Bigg/ \frac{\Gamma(v_n/2)|\kappa_n|^{-1/2}(v_0\sigma_0^2)^{v_0/2}}{\Gamma(v_0/2)|\kappa_0|^{-m/2}(v_n\sigma^2_n)^{v_n/2}}\frac{1}{(\pi)^{n/2}} \\
&= \frac{\Gamma(v_{n+1}/2)|\kappa_{n+1}|^{-1/2}}{(v_n\sigma^2_n+\frac{\kappa_n}{\kappa_{n+1}}(x_j-\boldsymbol{\beta}^\star x)^2)^{v_{n+1}/2}}\frac{(v_n\sigma^2_n)^{v_n/2}}{\Gamma(v_n/2)|\kappa_n|^{-1/2}}\frac{1}{\pi^{1/2}} \\
&= \frac{\Gamma((v_n+1)/2)}{\Gamma(v_n/2)}\Big(\frac{|\kappa_n|}{|\kappa_{n+1}|}\Big)^{1/2}\frac{(v_n\sigma^2_n)^{v_n/2}}{(v_n\sigma^2_n+\frac{\kappa_n}{\kappa_{n+1}}(x_j-\boldsymbol{\beta}^\star x)^2)^{(v_n+1)/2}}\frac{1}{\pi^{1/2}} \\
&= \frac{\Gamma((v_n+1)/2)}{\Gamma(v_n/2)}\Big(\frac{|\kappa_n|}{|\kappa_{n+1}|\pi}\Big)^{1/2}\frac{(v_n\sigma^2_n)^{v_n/2}}{(v_n\sigma^2_n)^{v_n/2}(v_n\sigma^2_n)^{1/2}} \\
&\times \frac{1}{(1+\frac{\kappa_n}{\kappa_{n+1}v_n\sigma^2_n}(x_j-\boldsymbol{\beta}^\star x)^2)^{(v_n+1)/2}} \\
&= \frac{\Gamma((v_n+1)/2)}{\Gamma(v_n/2)}\Big(\frac{|\kappa_n|}{|\kappa_{n+1}|v_n\sigma^2_n\pi}\Big)^{1/2}\Big(1+\frac{\kappa_n(x_j-\boldsymbol{\beta}^\star x)^2)}{\kappa_{n+1}v_n\sigma^2_n}\Big)^{-(v_n+1)/2} \\
&= \frac{\Gamma((v_n+1)/2)}{\Gamma(v_n/2)}\frac{1}{\sqrt{v_n\pi\frac{|\kappa_{n+1}|\sigma^2_n}{|\kappa_n|}}}\Big(1+\frac{1}{v_n}\frac{\kappa_n}{\kappa_{n+1}\sigma^2_n}(x_j-\boldsymbol{\beta}^\star x)^2\Big)^{-(v_n+1)/2}
\end{aligned}
\tag{A.10}
$$

A.1.3 Detailed derivation of a model with a known mean

The marginal likelihood

With the segment likelihood (3.29), the prior (3.30) and the mean $\mu = \boldsymbol{x}'_{1:n}$, the segment probability can be written like

$$
\begin{aligned}
P(\boldsymbol{x}_{1:n}|\beta, v_0, \sigma_0^2) &= P(\boldsymbol{x}_{1:n}|\boldsymbol{x}'_{1:n}, \sigma^2)p(\sigma^2|v_0, \sigma_0^2) \\
&\propto \mathcal{N}\left(\boldsymbol{x}_{1:n}|\boldsymbol{x}'_{1:n}, \sigma^2\right) \times \mathcal{X}^{-2}\left(\sigma^2|v_0, \sigma_0^2\right) \\
&= (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{x}'_i)^2} \\
&\times \Gamma(v_0/2)^{-1}\left(\sigma_0^2 v_0/2\right)^{v_0/2} \sigma^{-2(1+v_0)/2} e^{-\frac{1}{2\sigma^2}v_0\sigma_0^2} \\
&= \frac{1}{Z_{\mathcal{N}\mathcal{X}^{-2}}(v_0\sigma_0)} \sigma^{-2(1+v_0+n)/2} e^{-\frac{1}{2\sigma^2}(v_0\sigma_0^2 + \sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{x}'_i))^2)} \\
Z_{\mathcal{N}\mathcal{X}^{-2}}(v_0, \sigma_0) &= (2\pi)^{n/2}\Gamma(v_0/2)\left(\sigma_0^2 v_0/2\right)^{-v_0/2} \\
&,
\end{aligned}
\tag{A.11}
$$

where $Z_{\mathcal{NIX}}(v_0, \sigma_0)$ is the normalization constant of the distribution. With the comparison of the segment distribution with the normal inverse chi squared distribution with the updated variance like

$$
\begin{aligned}
p(\boldsymbol{x}_{1:n}|\boldsymbol{x}'_{1:n}, \sigma^2)p(\sigma^2|v_0, \sigma_0^2) &= p(\boldsymbol{x}_{1:n}|\boldsymbol{x}'_{1:n}, v_n, \sigma_n^2) \\
\frac{1}{Z_{\mathcal{N}\mathcal{X}^{-2}}(v_0\sigma_0)} \sigma^{-2(1+v_0+n)/2} e^{-\frac{1}{2\sigma^2}(v_0\sigma_0^2 + \sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{x}'_i)^2)} &= \frac{1}{Z_{\mathcal{N}\mathcal{X}^{-2}}(v_n\sigma_n)} \sigma^{-2(1+v_n)/2} e^{-\frac{1}{2\sigma^2}(v_n\sigma_n^2)}
\end{aligned}
\tag{A.12}
$$

, the hyperparameter updates can be written like

$$
\begin{aligned}
v_n &= v_0 + n \\
\sigma_n^2 &= \frac{1}{v_n}\left(v_0\sigma_0^2 + \sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{x}'_i)^2\right) .
\end{aligned}
\tag{A.13}
$$

With updated the hyperparameters v_n and σ_n^2, the posterior distribution of the variance σ can be formulated like

$$
p(\sigma^2|\boldsymbol{x}_{1:n}, \boldsymbol{x}'_{1:n}, v_0, \sigma_0^2) \propto \mathcal{N}\mathcal{X}^{-2}\left(\sigma^2|\boldsymbol{x}'_{1:n}, v_n, \sigma_n^2\right) . \tag{A.14}
$$

The posterior predictive

On the base of the marginal likelihood, the posterior predictive can be formulated like

$$\begin{aligned}
p(\boldsymbol{x}_{n+1}|\boldsymbol{x}_{1:n}) &= \frac{p(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{1:n})}{p(\boldsymbol{x}_{1:n})} \\
&= \frac{Z_{\mathcal{N}\mathcal{X}^{-2}}(v_{n+1}, \sigma^2_{n+1})}{(2\pi)^{n/2} Z_{\mathcal{X}^{-2}}(v_0, \sigma_0)} \frac{(2\pi)^{n/2} Z_{\mathcal{X}^{-2}}(v_0, \sigma_0)}{Z_{\mathcal{N}\mathcal{X}^{-2}}(v_n, \sigma_n)} \\
&= \frac{Z_{\mathcal{N}\mathcal{X}^{-2}}(v_{n+1}, \sigma^2_{n+1})}{Z_{\mathcal{N}\mathcal{X}^{-2}}(v_n, \sigma_n)} \\
&= \frac{(2\pi)^{n+1/2}\Gamma(v_{n+1}/2)\left(\sigma^2_{n+1}v_{n+1}/2\right)^{-v_{n+1}/2}}{(2\pi)^{n/2}\Gamma(v_n/2)\,(\sigma^2_n v_n/2)^{-v_n/2}} \\
&= \frac{\Gamma(\frac{v_n+1)}{2}}{\Gamma(\frac{v_n}{2})} \frac{1}{\sqrt{2\pi}} \left(\sigma^2_n v_n/2\right)^{-1/2} \left(\frac{\sigma^2_n v_n}{\sigma^2_{n+1} v_{n+1}}\right)^{-\frac{v_n+1}{2}} \\
&= \frac{\Gamma(\frac{v_n+1)}{2}}{\Gamma(\frac{v_n}{2})} \frac{1}{\sqrt{\pi\sigma^2_n v_n}} \left(\frac{v_{n+1}(\frac{v_n}{v_{n+1}}\sigma^2_n + \frac{1}{v_{n+1}}\left(\boldsymbol{x}_{n+1} - \boldsymbol{x}'_{1:n}\right)^2)}{v_n\sigma^2_n}\right)^{-\frac{v_n+1}{2}} \\
&= \frac{\Gamma(\frac{v_n+1)}{2}}{\Gamma(\frac{v_n}{2})} \frac{1}{\sqrt{\pi\sigma^2_n v_n}} \left(\frac{v_n\sigma^2_n + \left(\boldsymbol{x}_{n+1} - \boldsymbol{x}'_{1:n}\right)^2)}{v_n\sigma^2_n}\right)^{\frac{-v_n+1}{2}} \\
&= \frac{\Gamma(\frac{v_n)}{2}}{\Gamma(\frac{v_n+1}{2})} \frac{1}{\sqrt{\pi\sigma^2_n v_n}} \left(1 + \frac{\left(\boldsymbol{x}_i - \boldsymbol{x}'_{1:n}\right)^2)}{v_n\sigma^2_n}\right)^{-\frac{v_n+1}{2}} \\
&= t_{v_n}\left(\boldsymbol{x}_{n+1}|\boldsymbol{x}'_{1:n}, \sigma^2_n\right) \ ,
\end{aligned} \tag{A.15}$$

where t_{v_n} is the Student's t-distributed probability density like Eq. (A.16).

A.1.4 Student's t-distribution

As reference, the Student's t-distributed probability density formulation is written like

$$t_v(x|\mu, \sigma^2) = \frac{\Gamma(\frac{v}{2} + \frac{1}{2})}{\Gamma(\frac{v}{2})} \frac{1}{\sqrt{v\pi\sigma^2}} \left(1 + \frac{1}{v}\left(\frac{(x-\mu)^2}{\sigma^2}\right)\right)^{-\frac{v+1}{2}} \quad , \tag{A.16}$$

where $v > 0$ is the degrees of freedom, mu is the mean and $\sigma^2 > 0$ is the scale.

A.2 Scenarios

A.2.1 Three point motion

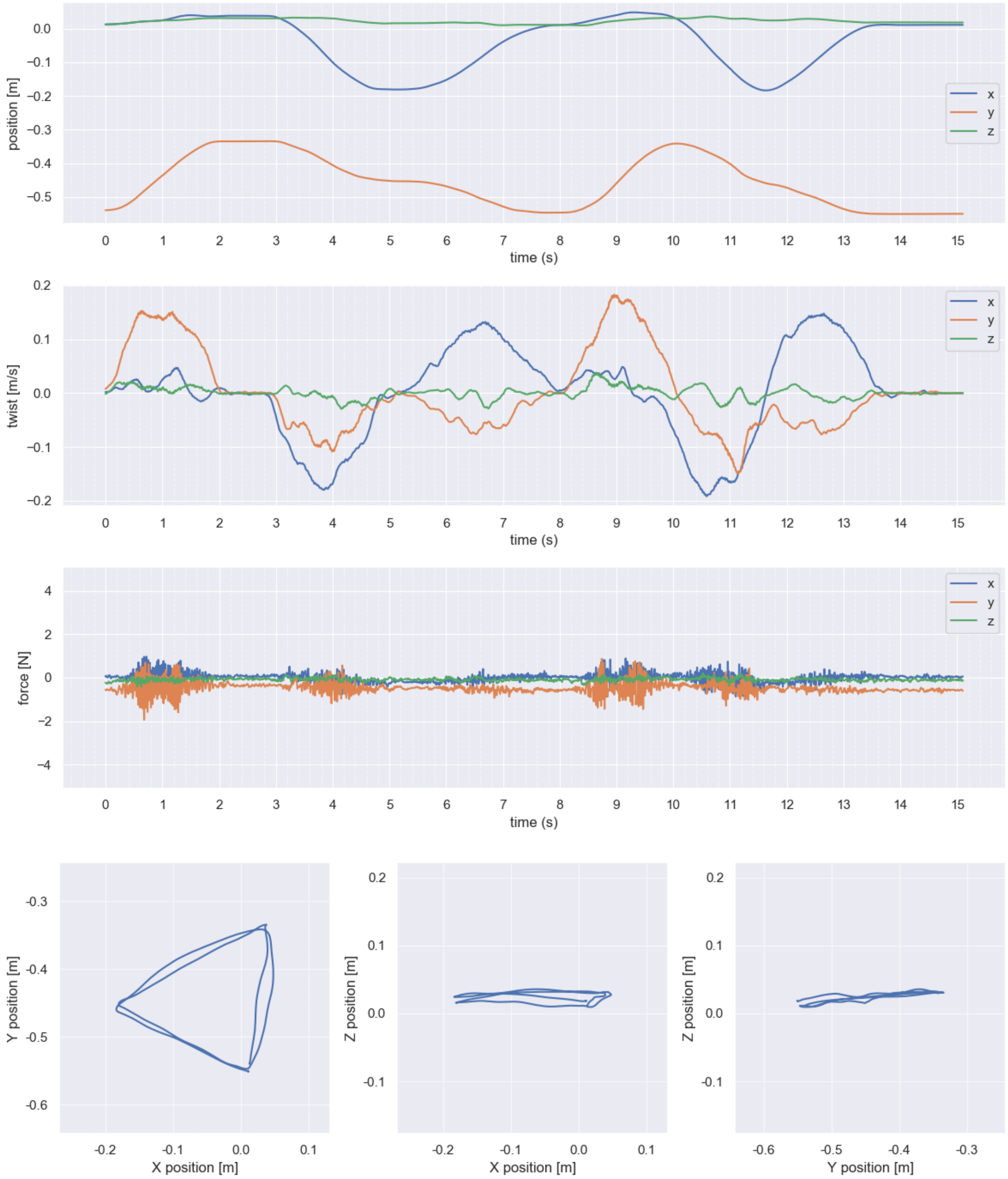


Figure A.1: Shows the acquired end effector position, twist and the environmental force of the Three Point Scenario

A.2.2 Parkour

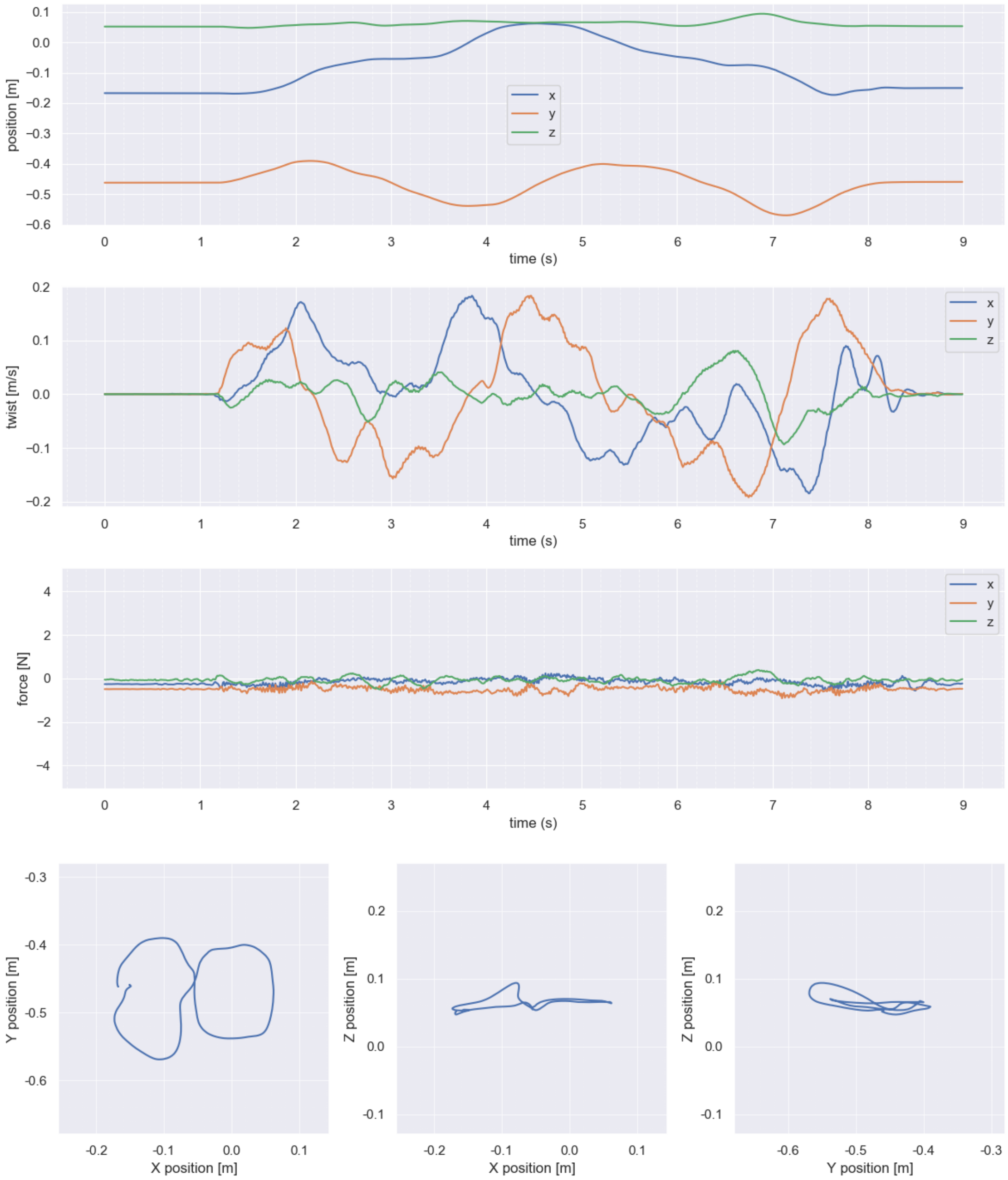


Figure A.2: Shows the acquired end effector position, twist and the environmental force of the parkour Scenario

A.2.3 Pick and place

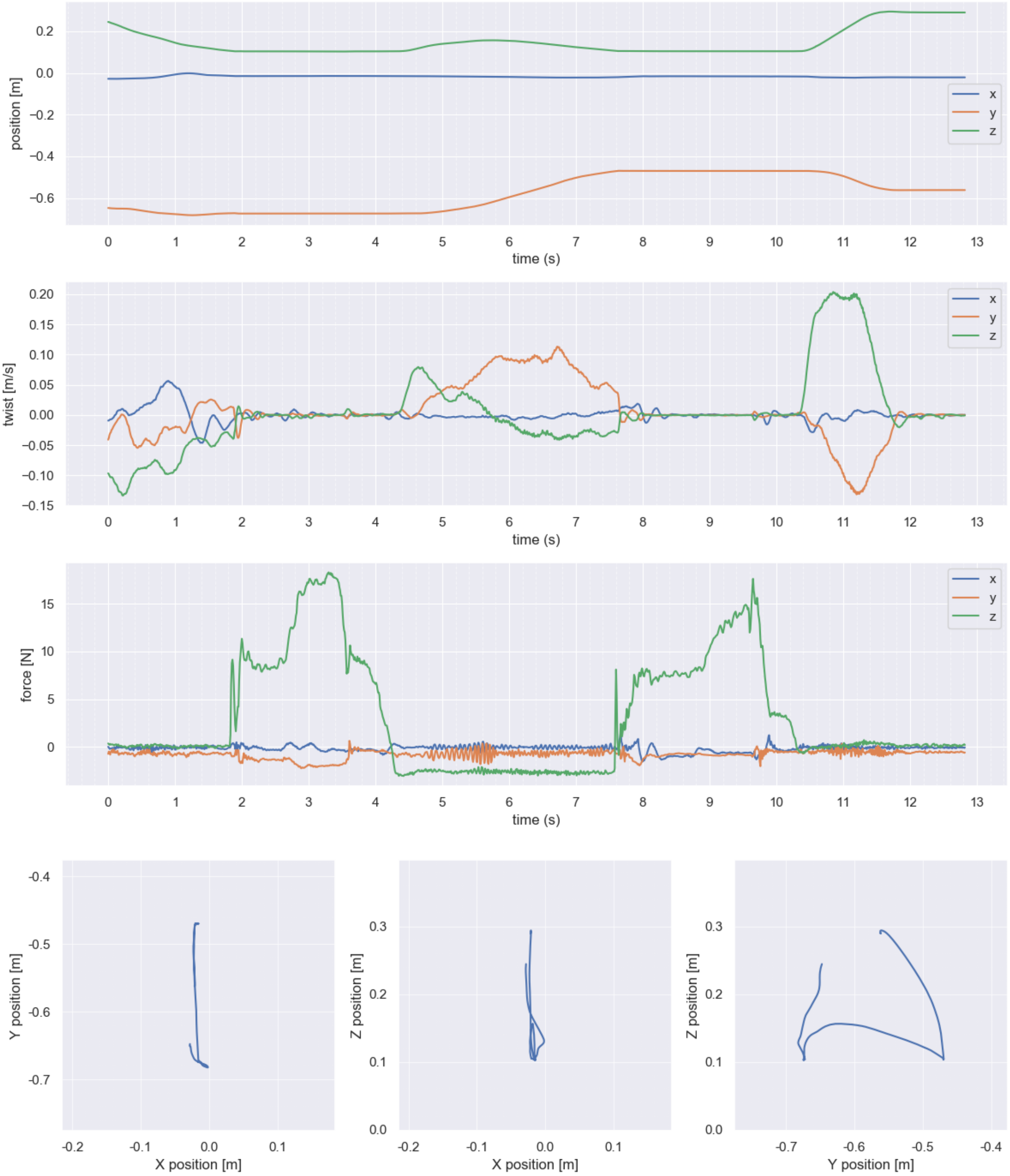


Figure A.3: Shows the acquired end effector position, twist and the environmental force of the Pick and Place Scenario

A.2.4 Soap Dispenser

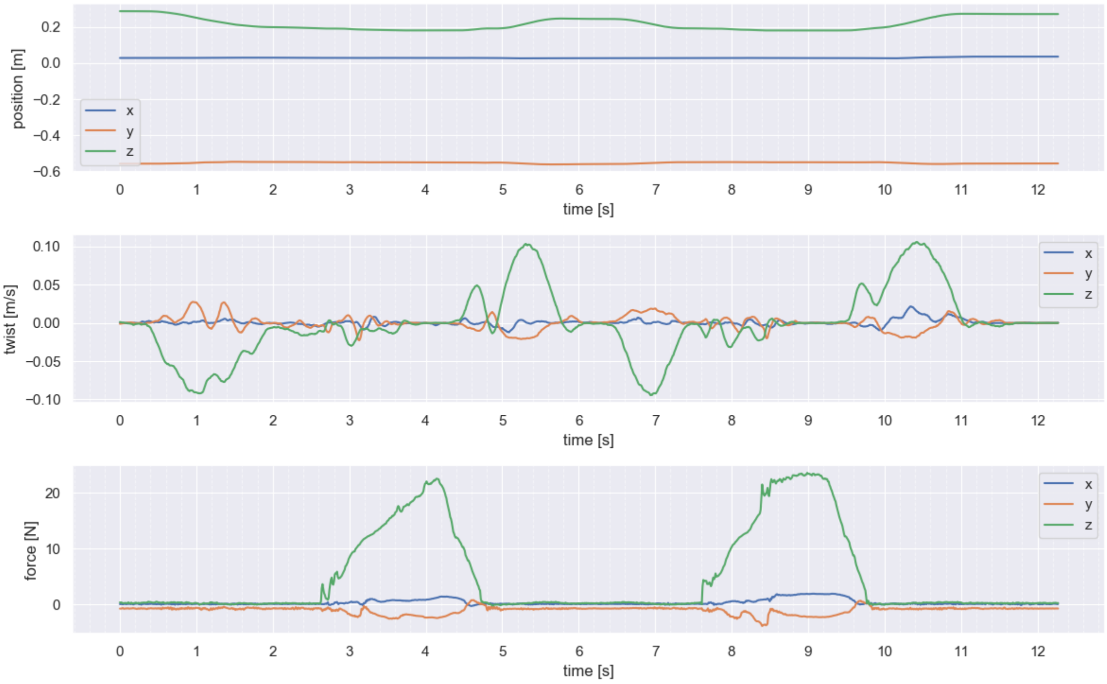


Figure A.4: Shows the acquired end effector position, twist and the environmental force of the Soap Dispense Scenario

List of Figures

Acronyms and Notations

DOF Degree of Freedom

BCPD Bayesian Changepoint detection

oBCPD online Bayesian Changepoint detection

GUI Graphical User interface

PPM Product Partition Model

FT-Sensor Force Torque Sensor

ZVC Zero velocity Crossing

Bibliography

[AOFH03] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger. Cartesian impedance control of redundant robots: recent results with the dlr-light-weight-arms. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 3704–3709 vol.3, Sep. 2003. doi:10.1109/ROBOT.2003.1242165.

[BH92a] Daniel Barry and J. A. Hartigan. Product partition models for change point problems. *Ann. Statist.*, 20(1):260–279, 03 1992. URL: https://doi.org/10.1214/aos/1176348521, doi:10.1214/aos/1176348521.

[BH92b] Daniel Barry and J. A. Hartigan. Product partition models for change point problems. *Ann. Statist.*, 20(1):260–279, 03 1992. URL: https://doi.org/10.1214/aos/1176348521, doi:10.1214/aos/1176348521.

[CB04] S. Calinon and A. Billard. Stochastic gesture production and recognition model for a humanoid robot. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2769–2774 vol.3, 2004.

[Dah09] David Dahl. Modal clustering in a class of product partition models. *Bayesian Analysis*, 4, 06 2009. doi:10.1214/09-BA409.

[Fea06] Paul Fearnhead. Exact and efficient bayesian inference for multiple changepoint. *Statistics and Computing*, 16:203–213, 06 2006. doi:10.1007/s11222-006-8450-8.

[FMJ02] Ajo Fod, Maja J. Mataric, and Odest Chadwicke Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12:39–54, 2002.

[JGPV17] Ibrahim Jasim GHALYAN, Peter Plapper, and Holger Voos. Contact-state modelling in force-controlled robotic peg-in-hole assembly processes of flexible objects using optimised gaussian mixtures. *Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture*, 231:1301–1308, 07 2017. doi:10.1177/0954405415598945.

[LKK16] J. F. Lin, M. Karg, and D. Kulic. Movement primitive segmentation for human motion modeling: A framework for analysis. *IEEE Transactions on Human-Machine Systems*, 46(3):325–339, June 2016. `doi:10.1109/THMS.2015.2493536`.

[LNMP17] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters. Learning movement primitive libraries through probabilistic segmentation. *International Journal of Robotics Research*, 36(8):879–894, July 2017.

[MKA13] Rakesh Malladi, Giridhar Kalamangalam, and Behnaam Aazhang. Online bayesian change point detection algorithms for segmentation of epileptic activity. pages 1833–1837, 11 2013. `doi:10.1109/ACSSC.2013.6810619`.

[MTS12] Franziska Meier, Evangelos Theodorou, and Stefan Schaal. Movement segmentation and recognition for imitation learning. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 761–769, La Palma, Canary Islands, Apr 2012. PMLR. URL: `http://proceedings.mlr.press/v22/meier12.html`.

[Pom00] Marc Pomplun. Evaluation metrics and results of human arm movement imitation. In *In Proc. of First IEEE-RAS International Conference on Humanoid Robots*, 2000.

[QI03] Fernando A. Quintana and Pilar L. Iglesias. Bayesian clustering and product partition models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 65(2):557–574, 2003. URL: `http://www.jstor.org/stable/3647521`.

[Rug13] Eric Ruggieri. A bayesian approach to detecting change points in climatic records. *International Journal of Climatology*, 33:520–528, 02 2013. `doi:10.1002/joc.3447`.

[SHP20] S. Haddadin S. Haddadin and S. Parusel. Franka Emika GmbH, 2020. URL: `www.franka.de`.

[SK16] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics.* Springer Publishing Company, Incorporated, 2nd edition, 2016.

[XM07] Xiang Xuan and Kevin Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 1055–1062, New York, NY, USA, 2007. ACM. URL: `http://doi.acm.org/10.1145/1273496.1273629`, `doi:10.1145/1273496.1273629`.

[YZL^{+}19] Ali Yacoub, Yuchen Zhao, Niels Lohse, Yee Mey Goh, Peter Kinnell, Pedro Ferreira, and Ella-Mae Hubbard. Symbolic-based recognition of contact states for learning assembly skills. *Frontiers in Robotics and AI*, 6, 10 2019. `doi:10.3389/frobt.2019.00099`.

[ZKJ^{+}13] T. Zhang, M. Karg, J.F.Lin, D.Kulic, and G. Venture. Imu based single stride identification of humans. In *2013 IEEE RO-MAN*, pages 220–225, 2013.

License