Robust Human Motion Prediction for Safe and Efficient Human-Robot Interaction

by

Przemyslaw A. Lasota

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Autonomous Systems

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Signature redacted

Author .....
Department of Aeronautics and Astronautics
Signature redacted May 23, 2019

Certified by .....
Prof. Julie Shah
Signature redacted Thesis Supervisor

Certified by .....
Prof. Rachid Alami
Signature redacted Thesis Committee Member

Certified by .....
Dr. Terrence Fong
Thesis Committee Member

Certified by ..... Signature redacted
Prof. David Mindell
Thesis Committee Member

Accepted by ..... Signature redacted
Sertac Karaman
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Robust Human Motion Prediction for Safe and Efficient Human-Robot Interaction

by

Przemyslaw A. Lasota

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Autonomous Systems

Abstract

From robotic co-workers in factories to assistive robots in homes, human-robot interaction (HRI) has the potential to revolutionize a large array of domains by enabling robotic assistance where it was previously not possible. Introducing robots into human-occupied domains, however, requires strong consideration for the safety and efficiency of the interaction. One particularly effective method of supporting safe an efficient human-robot interaction is through the use of human motion prediction. By predicting where a person might reach or walk toward in the upcoming moments, a robot can adjust its motions to proactively resolve motion conflicts and avoid impeding the person's movements.

Current approaches to human motion prediction, however, often lack the robustness required for real-world deployment. Many methods are designed for predicting specific types of tasks and motions, and do not necessarily generalize well to other domains. It is also possible that no single predictor is suitable for predicting motion in a given scenario, and that multiple predictors are needed. Due to these drawbacks, without expert knowledge in the field of human motion prediction, it is difficult to deploy prediction on real robotic systems.

Another key limitation of current human motion prediction approaches lies in deficiencies in partial trajectory alignment. Alignment of partially executed motions to a representative trajectory for a motion is a key enabling technology for many goal-based prediction methods. Current approaches of partial trajectory alignment, however, do not provide satisfactory alignments for many real-world trajectories. Specifically, due to reliance on Euclidean distance metrics, overlapping trajectory regions and temporary stops lead to large alignment errors.

In this thesis, I introduce two frameworks designed to improve the robustness of human motion prediction in order to facilitate its use for safe and efficient human-robot interaction. First, I introduce the Multiple-Predictor System (MPS), a data-driven approach that uses given task and motion data in order to synthesize a high performing predictor by automatically identifying informative prediction features and combining the strengths of complementary prediction methods. With the use of three

distinct human motion datasets, I show that using the MPS leads to lower prediction error in a variety of HRI scenarios, and allows for accurate prediction for a range of time horizons.

Second, in order to address the drawbacks of prior alignment techniques, I introduce the Bayesian ESTimator for Partial Trajectory Alignment (BEST-PTA). This Bayesian estimation framework uses a combination of optimization, supervised learning, and unsupervised learning components that are trained and synthesized based on a given set of example trajectories. Through an evaluation on three human motion datasets, I show that BEST-PTA reduces alignment error when compared to state-of-the-art baselines. Furthermore, I demonstrate that this improved alignment reduces human motion prediction error.

Lastly, in order to assess the utility of the developed methods for improving safety and efficiency in HRI, I introduce an integrated framework combining prediction with robot planning in time. I describe an implementation and evaluation of this framework on a real physical system. Through this demonstration, I show that the developed approach leads to automatically derived adaptive robot behavior. I show that the developed framework leads to improvements in quantitative metrics of safety and efficiency with the use of a simulated evaluation.

Prof. Julie Shah
Thesis Supervisor

Prof. Rachid Alami
Thesis Committee Member

Dr. Terrence Fong
Thesis Committee Member

Prof. David Mindell
Thesis Committee Member

Prof. Bradley Hayes
Thesis Reader

Prof. Leia Stirling
Thesis Reader

Acknowledgments

I would not have been able to get to where I am today and to finish my degree without the support of many fantastic people. First, I would like to thank my advisor, Prof. Julie Shah. Back when I first started at MIT, I was on the fence about sticking around for a PhD, and was planning to just grab my Master's degree and run. As Julie's student, it did not take long for me to change my mind. Right from the start of my graduate career, Julie was an outstanding mentor and supported me in pursuing research ideas that I was most passionate about. Being Julie's student and deciding to stay for a PhD opened up countless exciting opportunities and allowed me to grow and develop as a researcher. If Julie had not been my advisor, I would have missed out on all of these opportunities, so I will be forever grateful to her for her mentorship and support throughout my graduate career.

Next, I want to thank my committee members for their support and mentorship: Prof. Rachid Alami for his insights on human-aware motion and task planning that shaped and refined my thinking and helped me develop my thesis ideas; Dr. Terry Fong for making me take a step back and think about the big picture when thinking about my research problems and for being my NSTRF mentor, giving me the unique opportunity of collaborating with NASA researchers and engineers in the Intelligent Robotics Group; and Prof. David Mindell, for using his decades of experience on human-machine interaction to ask many key questions that I would not have considered, which greatly influenced my work. I also want to than my thesis readers, Prof. Bradley Hayes and Prof. Leia Stirling, for their help with improving my thesis.

I would also like to thank those who influenced me to pursue a graduate degree in the first place. I'd like to thank Prof. Haym Benaroya for introducing me to academic research when I was an undergraduate student at Rutgers University by letting me be a part of his research group. I would also like to thank Prof. Haim Baruh for first introducing me to robotics and automation by giving me the opportunity to be part of the NYCRI program under his mentorship.

Next, I would like to thank everyone in the Interactive Robotics Group for the many years of friendship and support. From simple things like our coffee breaks and group lunches to unforgettable, unique moments like everyone learning how to sing "happy birthday" in Polish for my birthday one year, having you all as my labmates had a huge impact on making my experience at MIT so great and memorable. I would especially like to thank Joseph Kim for all his support and being my "grad school buddy" as we went through the entire PhD process together, Vaibhav Unhelkar for the years of humor that could always cheer me up and for allowing me to jump in on the BMW project for an exciting and fruitful collaboration, and Ankit Shah for many research discussions over coffee that helped push my work along and for greatly improving my work-life balance by convincing me to take breaks and go sailing on the Charles River or go winter hiking during IAP.

I would like to thank my family. I want to thank my parents, Monika and Witold, for the many sacrifices they made throughout the years in order to allow me to pursue my education and become the person I am today. From uprooting their lives by moving to the USA 20 years ago, to supporting my goals and aspirations at every step of the way, I would not have been able to get to where I am today without them. I also want to thank my siblings: my brother Sebastian, for being the trail blazer through the American education system (both for me and my parents), and my sister Paula, for always knowing how to cheer me up and her words of encouragement when things got tough. I also want to thank all of my family in Poland, for shaping me into who I am today and supporting me throughout the years.

Finally, I would like to thank my wife, Christina for provided unyielding love and support for me throughout my entire academic career. She was always there for me to celebrate every success and to cheer me up when things got tough. I would not have been able to achieve what I did without her by my side, and I am grateful for having her in my life.

Funding

This thesis was funded by the NASA Space Technology Research Fellowship Program and the National Science Foundation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Interest in the field of human-robot interaction (HRI) has been growing at an accelerating rate, as a wide variety of fields and applications stand to benefit from its use. In recent years, there has been a significant push and interest toward introducing robots on factory floors (Figure 1-1) [74, 110, 29, 146, 36] in homes (Figure 1-2) [48, 35], and even as assistants on board the International Space Station [44, 15, 6]. Human-robot interaction in these and many other environments will require robots to exist in close proximity to people, falling under the umbrella of physical human-robot interaction (pHRI). Due to the possible dangers that robots can pose to humans, supporting this type of interaction requires the development of techniques for maintaining safety. Furthermore, the efficiency of the interaction must also be considered; a robot that constantly impedes a person's motion or takes inefficient paths can, at best, be a nuisance in an environment such as a home or a mall, and, at worst, nullify any economical benefit of its use in industrial and commercial settings.

Using the definitions introduced by De Luca and Flacco [24], safe physical human-robot interaction involves three distinct robot capabilities: safety, coexistence, and collaboration. According to the authors, the robot's ability to maintain safety, whether through physical safeguards or software capabilities, is the primary capability the robot must possess. The next capability for physical human-robot interaction is the ability to safely coexist with co-located humans, wherein the human and robot share the same physical space but are not directly coordinating their actions. The final

capability, collaboration, involves performing a complex task with direct human interaction and coordination. In light of these definitions, in this thesis, I consider human-robot interaction to be divided into two categories: **collaboration** and **coexistence**. [1]

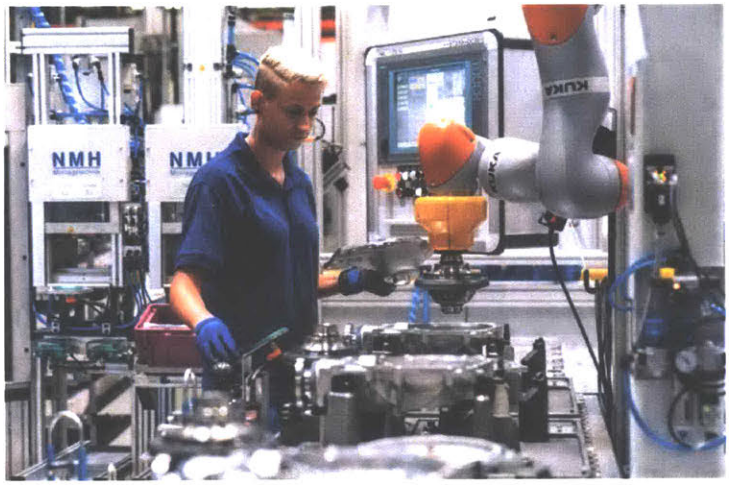


Figure 1-1: A human associate and a robot working together at an automotive factory. Image credit: BMW

As introduced above, safety is paramount for enabling close-proximity human-robot interaction. This is true regardless of whether the interaction involves collaboration or coexistence. Using the definitions introduced in [89], there are two main types of safety that need to be considered. The first is through direct physical contact. In simple terms, in order for HRI to be safe, no unintentional or unwanted contact can occur between the human and robot. Furthermore, if physical contact is required for a given task (or strict prevention of physical contact is neither possible nor practical) the forces exerted upon the human must remain below thresholds for physical discomfort or injury. This form of safety in HRI is defined as **physical safety**.

Preventing physical harm alone, however, does not necessarily translate to stress-free and comfortable interaction. Consider, for example, a hypothetical manufacturing scenario in which a robot uses a sharp cutting implement to perform a task in proximity to human workers, but is programmed to stop if a human gets too close.

[1] As the latter involves the human and robot navigating a shared space, I also refer to this type of interaction as **co-navigation**.

Figure 1-2: A home service robot interacting with a person. Image credit: Robosoft

While direct physical harm is prevented through careful programming, this type of interaction can be stressful for humans. Importantly, psychological discomfort or stress can also be induced by a robot's appearance, embodiment, gaze, speech, posture, and other attributes [108, 17]. Such stress can have serious negative effects on health [104], which makes stressful HRI a potential source of harm. Furthermore, psychological discomfort caused by any of the other aforementioned factors, as well as robotic violation of social conventions and norms during interaction, can also have serious negative effects on humans over time. We define the prevention of this type of indirect, psychological harm as maintaining **psychological safety**.

In support of safe and efficient human-robot interaction, researchers have proposed many unique methods and solutions. These include methods based on control, motion planning, considerations for psychological factors, and prediction. Human motion prediction, which is the focus of this thesis, is a particularly effective method for facilitating safe and efficient HRI in a wide variety of domains. Robot guides in museums, for example, can plan safer paths when co-navigating in potentially crowded spaces. Robots in homes can predict both ambulatory motions when traversing rooms as well as manipulation motions when directly assisting a person. Collaborative robots in factories can resolve motion conflicts with human co-workers, improving the work-

Figure 1-3: An artist's rendering of an Astrobee robot inside the International Space Station. Figure originally from [15].

ers' safety and comfort while simultaneously increasing the efficiency of the process. Space robots like Astrobee on the International Space Station [43] can understand where astronauts are moving toward to effectively co-navigate while carrying out its tasks.

While human motion prediction has the potential to improve safety and efficiency of human-robot interaction in many domains, current approaches possess several key limitations. First, current approaches to human motion prediction are often designed for specific types of tasks and motions and do not necessarily generalize well to other types of tasks; a predictor designed for a structured factory task, for example, will not work well for predicting the general motions of people walking through a corridor. While, for this particular example, it would be relatively easy to select the correct predictor, the decision will not always be clear-cut, and can be incredibly difficult for someone without expert knowledge in human motion prediction; for example, a predictor might work well for reaching but not for ambulatory motions. Furthermore, it is possible that no single predictor can provide accurate predictions by itself, and that some combination of several predictors might be required.

Another limiting factor of current methods of human motion prediction is caused by deficiencies in partial trajectory alignment, a tool that is often used in the context of prediction. Partial trajectory alignment involves observing a partially executed motion and mapping it to the corresponding section of a representative trajectory for

that given motion. This type of mapping is useful in goal-based human motion prediction, in which the target action or goal is predicted first, and then a representative trajectory of that action to project future motion. The main drawback of current approaches to partial trajectory alignment stems from the fact that they are often designed as alignment methods for generic time series and do not accommodate some qualities of human motion trajectories. Namely, the presence of overlapping regions and temporary stops can lead to large errors in alignment, which, in turn, can lead to large errors in prediction.

As a result of the drawbacks in generalizability of prior prediction methods and deficiencies in partial trajectory alignment, current human motion prediction approaches are not sufficiently robust [2] to be applied without requiring expertise in the field of human prediction and potentially tedious manual parameter tuning for each specific HRI scenario. This limits the use of human motion prediction in practice, hindering its use in supporting safety and efficiency in human-robot interaction. The goal of this thesis is to overcome these drawbacks by developing data-driven techniques capable of leveraging motion and task data from a given HRI scenario in order to produce robust and accurate predictions.

In Section 1.1, I propose a general motion prediction framework that, based on data from a given scenario, automatically synthesizes a high performing predictor from a library of candidate complementary predictors. I introduce three distinct human motion datasets used for the evaluations in this thesis, and discuss the key differences between these datasets that makes them suitable for assessing the robustness of the developed framework. I then show that my method results in improved prediction accuracy for all of the datasets when compared to baseline predictors. In Section 1.2, I present a novel method of partial trajectory alignment based on Bayesian estimation called the Bayesian Estimator for Partial Trajectory Alignment (BEST-PTA). I describe how this method specifically addresses the drawbacks of prior approaches, and demonstrate its robustness by showing that it leads to reduced alignment error

[2] As it can have different meanings in some fields of study, in this context, "robust" means "capable of performing without failure under a wide range of conditions" [122].

on all three human motion datasets when compared to baseline partial trajectory alignment methods. Furthermore, I demonstrate that the error of human motion prediction is reduced when my alignment method is used as part of my prediction framework. Finally, in Section 1.3 I describe the evaluation of an integrated prediction and planning system aimed at assessing the impact of the developed methods on the quality of human robot interaction. First, I describe a physical demonstration of this system, implemented in a test factory work cell, and discuss the key observations derived from comparing this system against baseline approaches. Next, I describe an evaluation performed in simulation designed to provide a quantitative assessment of key parameters of interaction and show that the developed system results in safer and more efficient interaction. Finally, I assess the impact of using BEST-PTA for alignment on these interaction parameters through a similar simulated evaluation.

1.1 Developing a Framework for Robust and Accurate Human Motion Prediction

As mentioned in the previous section, an effective method of supporting safety and efficiency in human-robot interaction is through robot adaptation based on prediction of human motion. By being able to anticipate where a person might be reaching or walking toward next, a robot can choose its actions or adjust its movement such that potential motion conflicts are avoided. Results from prior work indicate that the use of a human-aware motion planner, which avoids moving through locations of upcoming human occupancy, leads to more efficient teamwork, increased satisfaction with the robot, and higher perceived safety and comfort [88]. The ability to predict where a person will move to next is a key component of such an approach, motivating the development of accurate, real-time prediction of human motion.

A variety of human motion prediction techniques have been developed. One main group of these techniques are goal-based methods, where an estimate of the person's goal location or action is used to predict motion. These include methods using

techniques like Gaussian Mixture Models (GMMs) [102], Bayesian classification using multivariate Gaussian models [117], inverse optimal control [163] [103], growing hidden Markov models [34], conditional random fields [79], and Markov Decision Processes [115].

The second main group of methods for human motion prediction are approaches based on the study of natural human motion. Examples of such work include encoding skeletal motion patterns as HMMs [140], reasoning on features of natural walking motion [81], utilizing biomechanical indicators of turns [145], encoding the influence of other agents via deep reinforcement learning [21], and considering the effects of social grouping [124].

While a large variety of techniques have been developed, the majority of these approaches were designed and tuned for specific types of motions or tasks, and thus would not necessarily generalize well to prediction in other domains. For example, predictors based on action models would not work well when applied to loosely structured tasks, or methods that reason on features of natural walking motions might not work well for predicting the motion of an athlete during a game. This creates a barrier for utilizing human motion prediction, as it may not always be clear which approach is best suited for a specific task.

Furthermore, accurate predictions at both short and long time horizons are needed for supporting safety and efficiency. For example, the robot must know where a person will be in the short term in order to maintain effective collision avoidance, but also must know the human's long-term predicted path in order to plan an efficient motion toward its own goal. As prediction performance of a particular method can vary as a function of the prediction time horizon, it might even be possible that no single technique works well, and that a combination of approaches is required.

Goal

In order to address the above problems, the objective of Chapter 3 is to develop a robust framework for human motion prediction that can easily be applied to a wide variety of tasks and motions (i.e., the framework should be able to adapt to many dif-

ferent scenarios and produce accurate predictions in each of them). More specifically, the developed prediction capability should be robust to differences in human motion qualities, like whether the motion is a reaching or walking motion, or how fast the person is moving; task qualities, like whether the interaction involves collaboration or co-navigation, or how long a typical motion lasts; and sensing qualities, like if the sensors are onboard or external, or how much tracking noise is present. Furthermore, the method should provide accurate predictions across a range of time horizons of interest.

To properly assess whether these goals are met, I introduce three distinct human motion datasets: reaching motions from a tabletop factory task, walking motions from an automotive assembly task, and simulated space motions designed to emulate an astronaut's motion in a microgravity environment. These datasets differ in all of the abovementioned key qualities (as well as several others), making them suitable for the evaluation of robustness.

Approach

In support of achieving the goal of creating a robust framework for human motion prediction, I introduce the Multiple-Predictor System (MPS). This framework is a data-driven approach to human motion prediction that, based on a variety of data encoding how a person moves within a shared workspace and how he or she performs tasks, automatically selects a favorable combination of prediction methods to accurately predict human occupancy at various future time frames.

The combined predictor is synthesized in two steps. First, the parameters of the individual prediction methods are learned with the use of a training set. Next, using a second subset of data, the performance of each predictor is evaluated at various time horizons of interest for the particular task. The relative performances are analyzed with the use of the Polynomial Weights algorithm, a method for combining outputs of multiple experts. This step produces a set of weights for each individual predictor at each of the time horizons of interest. These weights are then used to select which predictor should be used at each time horizon.

In a proof-of-concept implementation of this framework, I incorporate three prediction techniques: a velocity-based position projection method, a time series classification method, and a sequence prediction method. The velocity-based position projection method estimates the person's current velocity with the use of a Savitzky-Golay filter [130] and predicts a straight-line motion at that velocity. The time series classification method uses the Bayesian classification approach introduced in [117] for goal prediction, and is extended for motion prediction through partial trajectory alignment to the mean trajectory of a given action. Finally, the sequence prediction method finds patterns in action sequences with the use of the algorithm introduced in [93], and is extended for motion prediction in the same manner as the time series classification method.

These specific predictors were implemented due to their complementary nature in several key qualities. The velocity-based position projection method and time series classification methods, for example, both reason on motion data, while the sequence prediction method considers only the sequence in which actions are done. Another example is that the time series classification and sequence prediction methods are goal-based, while the velocity-based method does not utilize a concept of goal locations or discrete sets of actions. These complementary attributes make these methods a suitable set of individual predictors for the MPS.

Results and Contributions

In order to determine the robustness of the MPS to the different types of tasks and motions present in the aforementioned human motion datasets, I carried out a leave-one-out cross validation for each of the datasets. To test robustness to differences in additional qualities beyond the ones already present between the datasets, additional dataset variants were generated by manipulating parameters such as the number and sequence of actions. The results of the cross-validation indicate that the MPS is robust to all of the considered parameters, with the MPS outperforming the individual prediction methods for all datasets and the derived variants.

Besides reduction in mean prediction error, the evaluation also confirmed that the MPS learns to exploit the complementary nature of the candidate predictors. Namely, I show that the MPS intelligently down-selects to two of the three predictors based on the given scenario. For example, when the order of the sequences is randomized, the MPS automatically synthesizes a predictor that does not involve the sequence prediction method. Besides down-selecting, the MPS also decides what predictor should be used at each time horizon. For example, an intuitive result that is derived from the framework is that the velocity-based method should be used for short time horizons, while the other two methods should be selected at longer time horizons. The time horizon at which this switch should happen, however, varies by a large margin between some datasets, highlighting the utility of the data-driven learning process.

In summary, I developed a novel human motion prediction framework that uses data from a given task to synthesize a high performing predictor from a set of complementary methods based on their relative performances at various time horizons. Using an evaluation involving three distinct datasets, I show that the developed method is robust to a variety of task, motion, and sensing parameters, in that it outperforms the individual predictors in all tested scenarios. Through an analysis of the composition of the synthesized predictors, I demonstrate that the framework adapts to changes in key task and motion parameters automatically and learns to intelligently select appropriate predictors for each prediction time horizon.

1.2 Improving Robustness of Human Motion Prediction via a Bayesian Estimator for Partial Trajectory Alignment

Temporal alignment of time series data is a problem under investigation across a wide variety of domains and tasks, including phenome alignment in speech processing [61], subsequence matching in medical data [157], and audio alignment in musical performance [28]. Within robotics-related applications — specifically, the field of

human-robot interaction — human motion trajectories represent a type of time series that is of particular interest. Temporal alignment of such trajectories is a common problem in activity recognition [57, 149], action and motion prediction [91, 117, 31], and gesture recognition [10, 143]. These techniques allow robots to understand and predict human intentions, which can be leveraged to improve the safety and efficiency of human-robot interaction.

One of the most foundational approaches to time series alignment is Dynamic Time Warping (DTW) [128]. This method, originally developed for speech recognition, uses dynamic programming to identify an optimal time alignment that minimizes the distance between the aligned signals. Several variants of DTW have emerged since its introduction, including FastDTW [129], which provides an approximate solution to DTW with linear time and space complexity (as opposed to the quadratic complexity of DTW), and derivative dynamic time warping (DDTW) [68], which aligns signals with respect to their first derivative in order to achieve more-intuitive alignments between signals in which a feature (e.g., a peak) is more pronounced in one signal over the other. More recently, researchers developed generalized time warping (GTW) [162], a technique specifically geared toward the alignment of human motion obtained from multi-modal sources (such as video, motion capture, and accelerometer data).

While these methods of full time series alignment are useful in many applications, in the context of prediction, the ability to align partially observed trajectories is necessary. Distinct from temporally aligning two complete time series, online partial trajectory alignment involves observing streaming trajectory data and, as each new position is collected, identifying a suitable alignment between the latest point of the partial trajectory and a point along a reference trajectory for that motion. This capability is particularly useful for early prediction of motions, actions, or gestures, as only part of a trajectory is available in these situations, precluding the use of techniques designed for full trajectory alignment.

As the abovementioned approaches are designed for full trajectory alignment, they are generally not well suited for partial alignment. Consequently, researchers have developed other techniques to address temporal misalignment with partial trajectory

observations. The approach used by Hayes and Shah [57], for example, does not search for an explicit temporal alignment, but instead divides trajectories into overlapping temporal segments over which to train independent Gaussian mixture models for activity recognition, and then uses a modified form of max pooling over a time window. Dong and Williams, on the other hand, developed a method that identifies the best alignment between the current state and trained probabilistic flow tubes (PFTs) based on the distance between the current state and the PFT, as well as the elapsed time in the partial execution [31]. The framework introduced in [91] also uses a metric of alignment based on distance and time, but it involves searching for the nearest point in a temporal moving window.

Besides the above techniques built into prediction and recognition frameworks, researchers have also developed techniques specifically for partial time series alignment. In one such work, the authors developed an online variant of DTW (O-DTW) [28] in which the entries of the time warp matrix are computed iteratively as new data is received, and the direction of the search is guided by the location of the current best alignment value. The number of computed values at each iteration is bounded, resulting in constant time complexity. The path cost calculations incorporate the standard DTW formulation but are restricted to the previously computed values, allowing for online computation of an optimal alignment. Building upon this work, the authors of [99] created windowed time warping, which, by dividing the overall alignment problem into sub-alignments on smaller, overlapping regions, achieves accuracy comparable to O-DTW while improving run time.

Latecki et al presented another approach, optimal sequence bijection (OSB) [92], which focused on creating a one-to-one mapping (bijection) between sequences and effectively handling trajectories containing outliers. The resilience to outliers is implemented by allowing for the skipping of points in both the partial and full trajectories during alignment, and by incorporating a *jump cost* that penalizes such skips proportionally to the number of points skipped in a row. To achieve linear time complexity, this approach includes a warping window size parameter and limits the maximum number of points the algorithm can skip in a row.

While the above methods achieved success in partial time series alignment, aligning human motion trajectories poses unique challenges that can adversely affect these methods' performance. Specifically, trajectory overlap and temporary stops can be problematic, as the above methods' distance metrics are generally described in terms of Euclidean norm. This is an issue because trajectories with these features will have groups of points near each other in Euclidean space but not temporally, causing large errors in alignment. When combined with variability in a given trajectory's execution speed, it becomes difficult to set the above methods' parameters such that they can account for all of these qualities and still produce satisfactory results. Furthermore, it is desirable to output a distribution over possible correspondence points, but the above techniques do not provide this capability.

Goal

As discussed above, previous methods of online partial time series alignment can perform poorly when applied to trajectory alignment, especially with trajectories containing overlapping regions and temporary stops, which are present in many real-world trajectories. Therefore, the goal of Chapter 3 is to develop a method of online partial time series alignment specifically geared toward such trajectories. The developed method needs to accommodate both stops and overlaps in order to provide a better estimate of the progress through a trajectory than prior approaches. In order to characterize the uncertainty over the alignment, the method also needs to output a distribution over the potential alignment points.

As this technique is being developed in the context of improving the robustness of human motion prediction, the developed method also needs to accommodate a variety of different types of motions. This involves testing the method on both manipulation and ambulatory motions with different numbers of degrees of freedom, accelerations, and speeds. Furthermore, in order to verify its utility for use in prediction, another goal is to demonstrate that the use of this method leads to reduced human motion prediction error.

Approach

In order to address the drawbacks of prior partial trajectory alignment techniques, I introduce BEST-PTA (Bayesian ESTimator for Partial Trajectory Alignment), a Bayesian estimation framework composed of a mixture of optimization, supervised learning, and unsupervised learning components that are trained and synthesized based on a given set of example trajectories. Similarly to the MPS, BEST-PTA leverages data from a given task in order to construct a high performing aligner.

As part of this framework, I address the problem of trajectory overlap by introducing an optimization-based technique for trajectory segmentation that divides a trajectory such that there is minimal within-segment overlap. The segmentation, along with a trained segment classifier, are then used to guide the partial trajectory alignment away from overlapping regions at a different time in the trajectory. In order to track progress through these segments, I also introduce a state machine with Bayesian transitions based on segment duration priors and segment classifier likelihoods.

I address the issue of temporary stops in motion by developing a stop detection algorithm based on a Dirichlet Process Gaussian Mixture Model fit to speeds observed in the trajectory. Once the stops are recognized, the framework identifies them as separate segments and learns a different set of parameters for these special regions. As temporary stops can lead to poor alignment even for the alignment of full trajectories, I also introduce a novel approach to the alignment of such trajectories, producing ground truth alignments that more accurately convey progress through a trajectory. The approach is based on matching the stop segments of the two trajectories and utilizing a linear temporal interpolation in these regions while using DTW elsewhere.

In order to produce a distribution over alignment points, the framework uses temporal priors and distance-based likelihoods. The temporal correspondence priors are learned by modelling typical time index alignments observed in the training set with the use of a Gaussian kernel distribution fit. The likelihoods are constructed based on the distribution of distances between aligned points observed in training.

During execution, as each new position in the partial trajectory is received, the prior and likelihood are combined to arrive at a Bayesian estimate of the overall distribution over temporal alignments.

Results and Contributions

I evaluated the developed framework against three partial time series alignment baselines: O-DTW, OSB, and a moving window approach (which were briefly introduced in the introduction of this section). In order to verify the robustness of the approach, these tests were performed on the three distinct human motion datasets. A leave-one-out cross-validation was performed on each of these datasets. As the motions in the datasets contained both stop segments and overlaps, dataset variants were created in which the stop segments were removed in order to test the technique on motions with overlaps but without stops.

The results of the evaluations show that BEST-PTA outperforms each of the baseline methods on all datasets and their variants. This indicates that BEST-PTA is more robust to different types of motions and the presence of stops and overlaps than the baseline methods. As the alignment is to be used for human motion prediction, large errors are particularly undesirable, as these can translate to large prediction errors. Through the evaluation, I also show that besides reducing the mean alignment error, BEST-PTA also leads to fewer such large errors.

In order to demonstrate the benefit of this improved alignment in the context of human motion prediction, I performed an additional analysis where BEST-PTA and the other partial trajectory alignment baselines were used for prediction as part of the Multiple-Predictor System introduced in Section 1.1. The results indicate that using BEST-PTA for partial trajectory alignment leads to lower mean prediction error for all three datasets. Furthermore, it also leads to fewer large prediction errors. This result provides support for the utility of BEST-PTA in providing robust and accurate predictions of human motion.

Overall, the main contribution of Chapter 4 was the development of a novel method for partial trajectory alignment based on Bayesian estimation with temporal

priors and distance-based likelihoods. I show that this method is robust to a variety of types of motions as well as to the presence of trajectory overlap and temporary stops. I also demonstrate that using my method leads to improvements in human motion prediction, both in terms of lower mean prediction error and the reduced incidence of large errors.

1.3 Evaluation of Safety and Efficiency of Human-Robot Interaction

In the previous two sections, I summarized the chapters of this thesis dedicated to improving the robustness and accuracy of human motion prediction. As the overarching objective of this thesis is to improve the utility of motion prediction in order to benefit the safety and efficiency of human-robot interaction, the obtained improvements in prediction accuracy and partial trajectory alignment cannot be viewed in isolation. In order to understand the effects of the developed methods on safety and efficiency, there is a need to integrate the output of the prediction with an appropriate robot planner and apply this integrated system to an HRI scenario.

In order to take advantage of predictions made at a range of future time horizons, it is necessary for the planner to include an explicit representation of time (i.e., perform planning in time). Without an explicit representation of time (i.e., treating the entire predicted human path as a static obstacle), planners are likely to return inefficient solutions, or might not be able to find a solution at all. For example, if the predicted path of a person goes in front of a robot constrained to a rail and the planner is trying to find a solution to move the robot to the other side of the rail, a planner that does not plan in time will fail to find a solution, even if the person will be in the robot's path for only an instant.

For collaborative robots, the need for human-aware task and motion planning has become evident [80]. Sisbot et al. provided one of the first human-aware motion planners for robot navigation by explicitly accounting for human preferences [134].

Recently, several factors have been incorporated for human-aware planning, including gaze [96], legibility [33], and proxemics [30, 105].

While the above approaches take human presence into account, they do not incorporate human motion prediction. Toward this goal, work by Ziebart et al, focused on prediction of walking motions using maximum entropy inverse optimal control [163]. Park et al, on the other hand, proposed predicting short term motions as a Sparse Pseudo-input Gaussian Process as part of an overall prediction framework [115]. Both of these works, however, use time-independent planning instead of planning in time.

Recently, two approaches for efficient path planning with explicit modeling of time have been explored. In their work, Khambhaita et al. first generated the global plan of the robot's motion without considering human motion, and then modified the execution of this path using timed elastic bands [69]. While considering the timing of the predicted human motion though the developed cooperative planner, this approach does not provide any guarantees with regard to path optimality, which is important for considerations of efficiency. An alternate approach that does provide such guarantees is to pose planning in time as a graph search problem [78]. In their work, Phillips et al. [118] provided the safe interval path planner (SIPP), a computationally attractive approach for this problem while maintaining the optimality of the resulting plan. Evaluation of SIPP assumed perfect prediction of the environment, and thus involved execution of a single plan generated prior to motion execution. However, in practice, predictions change during execution, requiring approaches for both online replanning and the interleaving of execution with planning.

Goal

The goal of Chapter 5 is to assess the effect of the developed methods for human motion prediction and partial trajectory alignment on the quality of human-robot interaction in terms of metrics of safety and efficiency. Toward this goal, the first objective is to create an integrated system that combines the methods for human motion prediction and partial trajectory alignment described in the previous sections with an effective robot planner. As discussed above, the planner requires the ability

to generate optimal plans with explicit representation of time and incorporate noisy, time-varying predictions.

Next, in order to provide support for the use of the developed techniques for use in real-world robotic systems, the second goal is to provide a demonstration of an HRI scenario with physical hardware with the use of the integrated prediction and planning framework. Besides serving as a proof-of-concept, the goal of the physical demonstration is to provide a qualitative assessment of the integrated system's impact on the quality of the interaction.

Finally, the last objective is to perform a quantitative analysis in simulation in order to assess the effect of the integrated system on the safety and efficiency of human-robot interaction. This includes a comparison of using the overall prediction and planning integration compared to other baseline co-navigation approaches and the effects of using BEST-PTA in comparison to prior alignment methods. The goal of this analysis is to provide an additional level of assessment that is not possible by only comparing the relative performances of the individual methods.

Approach

For the purpose of assessing the effect of using the prediction and alignment methods developed in this thesis on the quality of human-robot interaction, I focus on an automotive assembly scenario involving a robot delivering parts to workstations by moving along an industrial rail. In order to construct an integrated system containing prediction and planning, I combine prediction from the MPS with a robot planning framework based on the safe interval path planner. In order to allow for time-varying, noisy predictions, the framework includes additional components for using SIPP in a manner that accommodates interleaving planning and execution by monitoring the robot task progress and incoming human predictions.

The physical demonstration of the integrated system was developed in a test factory work cell using a human-safe robot arm moving along a linear rail, both of which are certified and used in industrial applications. Using human motion data collected with an RGB-D sensor, the MPS was trained in order to predict newly

observed human motions. During execution, predictions from the MPS were provided to the SIPP-based planner in order to plan the robot's motion. The behavior of the integrated system was compared to that of other baseline co-navigation methods.

The quantitative assessments of the developed methods were performed through simulation of a representative automotive factory work cell layout. In this work cell, the robot assists multiple stationary workers while a human worker periodically walks through the area, intersecting the robot's linear rail. Real human motion data was collected for simulating the motion of the mobile worker and training the prediction and alignment components. Two sets of simulated evaluations were performed: one to test the overall integrated prediction and planning system against previous approaches to co-navigation, and one for comparing BEST-PTA against baseline partial trajectory alignment methods.

Results and Contributions

The physical demonstration of the integrated prediction and planning framework was successfully carried out, demonstrating the feasibility of using the MPS for online prediction of human motion. The demonstration showed that by using a framework combining prediction with planning in time, the robot exhibits anticipatory behaviors that support safe and efficient interaction. Importantly, these behaviors are derived automatically, without the need to pre-define them in advance.

The results of the first set of simulated evaluations demonstrate that using the integrated prediction and planning framework outperforms prior approaches for human-robot co-navigation in factories. The second set of simulations shows that using BEST-PTA for partial trajectory alignment leads to improvements in safety and efficiency of human-robot interaction when compared to using baseline alignment methods.

In summary, in Chapter 5 I developed an integrated prediction and planning framework capable of accommodating time-varying predictions. Using this integrated framework, I was able to demonstrate the feasibility of using the prediction methods developed in this thesis on a real, physical robotic system. This is the

first robot system combining time-optimal path planning in time with output from a multiple-predictor framework demonstrated on factory-certified hardware. Furthermore, through a series of simulated evaluations, I was able to show that the prediction and partial trajectory alignment methods developed in this thesis lead to improvements in safety and efficiency of human-robot interaction.

Chapter 2

Background

2.1 Introduction

As outlined in Chapter 1, the aim of this thesis is to develop a robust method of human motion prediction in order to facilitate its widespread use in human-robot interaction. Furthermore, the objective is also to show that developing such prediction capability can improve the safety and efficiency of the interaction. In order to motivate the use of prediction for this goal, it is useful to place it in context of other methods of improving safety and efficiency in HRI.

The first purpose of this chapter, therefore, is to introduce these alternate, non-prediction methods and highlight their benefits and drawbacks in comparison to methods that do utilize prediction. I first discuss control-based methods, which can be split into pre- and post-collision control. The former category deals with control methods prior to contact between a human and robot. This involves limiting key parameters, such as velocity or energy, or preventing collisions from occurring by defining safety regions, tracking separation distance, and guiding robot motion away from humans. The post-collision sub-category involves techniques such as minimizing injury by switching between various control methods when a collision is detected, distinguishing between intentional and non-intentional contact, and allowing for safe physical contact when it is needed during collaboration.

Next, I highlight work focused on planning safer robot paths and motions in order to avoid potential collisions. By taking various human-related parameters such as separation distance or gaze direction directly into account when forming motion plans, a robot is able to choose safer and more efficient paths and motions.

In the following section, I describe methods of assessing and maintaining psychological safety during HRI. As mentioned in Chapter 1, maintaining psychological safety involves ensuring that interactions with a robot are stress-free and comfortable. Work in this field has included the development of metrics through physiological sensing, questionnaires, and behavioral metrics and identifying which factors — such as a robot's size and speed or a human's prior experience with robots — can affect perceived safety and comfort.

Next, I introduce prior work on the various techniques involving prediction in human-robot interaction. First, I briefly discuss work on human prediction of robots' motions and actions via explicit and implicit queues. Next, I introduce prior work on human activity prediction, including predicting future human actions based on low-level features derived from observing humans via sensors (such as cameras or motion-capture systems) as well as techniques that reason on sequences of actions. Finally, I provide an in-depth discussion of prior work in the field of human motion prediction, which is the focus of this thesis. Namely, I discuss work in two broad categories: predicting human motion based on inferring a person's goal and by utilizing characteristics of natural human motion. The former category focuses on predicting a goal action or location a person is moving toward and combining this prediction with a motion model to predict how the person will move toward this goal. The latter, category, on the other hand, does not consider such high-level goals, but instead focuses on studying qualities of natural human movement that can be used for prediction.

Finally, in the last section of this chapter, I discuss the merits of using human motion prediction in the context of enabling safe and efficient human-robot interaction. Namely, I compare the various sets of techniques highlighted in this chapter and discuss their benefits and drawbacks in the context of HRI and discuss the major advantages of human motion prediction, specifically. I also highlight some of the po-

tential shortcomings of these human motion prediction techniques, which were briefly introduced in Chapter 1.

2.2 Control-Based Methods in HRI

One common method for achieving safety during human-robot interaction is through low-level control of robot motion. This type of safety provision is often the simplest method of enabling safe human-robot coexistence, as it does not require complex prediction models or planners — nor, in some cases, does it even require sensing to monitor the human. Nonetheless, implementation of these solutions can be quite complex, as this method often includes time-critical constraints that require rapid execution.

Control methods for improving safety are divisible into two main categories: pre- and post-collision. Pre-collision control methods are implemented before human-robot collision occurs, either by ensuring collision does not occur in the first place or by bounding robot parameters such as velocity or energy. If unexpected or unpreventable contact occurs, post-collision control methods are designed to quickly detect the collision and minimize harm to both the human and robot. (Note that in this context, "collision" is not limited to blunt impacts, but can also include other harmful forms of contact, such as shearing, cutting, or puncturing.)

2.2.1 Pre-Collision Methods

Pre-collision control methods, sometimes referred to as "prevention" methods, are techniques intended to ensure safety during HRI by monitoring either the human, the robot, or both and modifying robot control parameters prior to incidence of collision or contact. The various techniques and methods designed to provide safety through pre-collision control discussed in this section are depicted in Figure 2-1 below.

One major subset of this category is focused on providing quantitative guarantees that a robot cannot pose any threat to a human, even in the event of a collision. This can be achieved by limiting a variety of parameters, such as the robot's joint velocity,

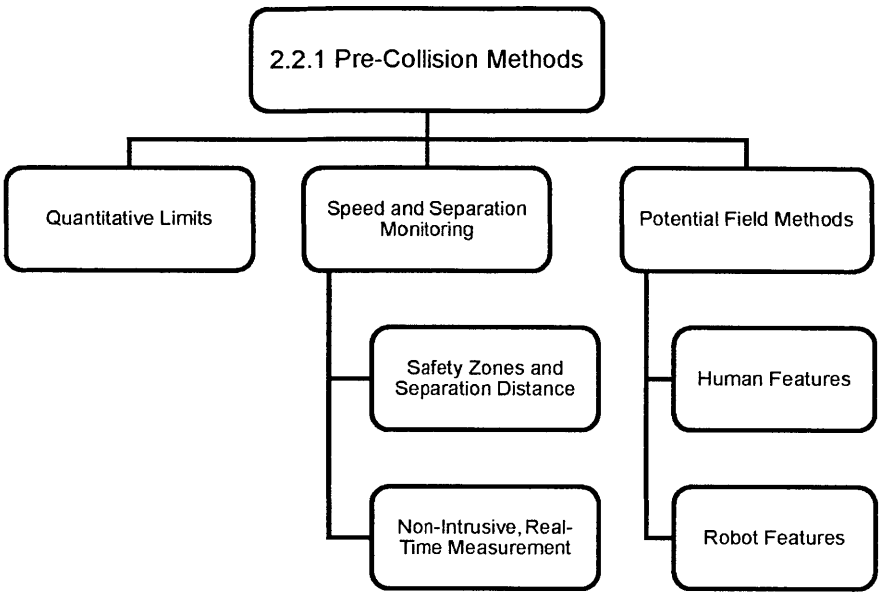


Figure 2-1: Diagram depicting the pre-collision control methods discussed in Section 2.2.1.

energy, or potential exertion of force. In [14], for example, the authors developed a trajectory planner that limits jerk, acceleration, and velocity. Other work focused on real-time tracking and limiting of the total amount of energy stored in the system, namely the sum of kinetic, gravitational potential, and elastic potential energies [86]. In [58], on the other hand, the authors developed a control approach that limits potential impact forces with static obstacles by imposing a safety envelope on torque commands of a position control algorithm. The presented method successfully limits the impact forces regardless of where on the robot the collision occurs. In [50], the authors took the unique approach of embedding injury knowledge into robot control by studying the relationship between robot mass, velocity, and impact geometry with injury to derive risk curves that characterize these relationships. The various methods of calculating and limiting velocities, potential impact forces, and energy allows for the abovementioned systems to provide quantitative guarantees about the robot's ability to inflict harm in the case of unwanted collision.

Another method of preventing collision through control is by slowing down or stopping the robot based on safety zones or distance of separation. The robotics and automation company ABB, for example, has developed a system called SafeMove,

which utilizes programmable, complex safety zones that can be used to control robot speed [76]. In contrast with static, predefined safety zones, the authors of [151] developed a system that incorporates dynamically changing zones based on robot joint positions and velocities and displays these zones on the surface around the robot via a projector. Moving beyond large safety zones and simply stopping the robot, in [90] the authors developed a safety system for close-proximity interaction with standard industrial robots that rapidly calculates the distance of separation between the human and the robot to gradually decrease the robot's speed according to a tunable function that can be adjusted based on task-dependent parameters. To ensure such a system does not overly hinder productivity, another approach is to take advantage of redundant degrees of freedom to simultaneously maintain safety and allow the robot to retain most of its speed by moving the robot's joints away from the human [161].

One significant challenge of deploying systems such as the ones cited above is providing nonintrusive methods of accurate human localization. One viable method presented in [125] uses sensor fusion techniques to combine data collected from stereo and range cameras in order to monitor an industrial workcell. The system detects people and robots within the environment and generates dynamically changing "danger zones" based on the position and trajectory of the robot. In contrast, the authors of [42] utilized depth sensors to estimate the distance between the robot and both static and moving obstacles. This real-time distance measurement, as well as an estimate of obstacle velocity, was then used with a controller based on repulsive force vectors as a collision prevention technique.

To reduce the possibility of occlusion and to accommodate unstructured tasks for which the optimal sensing locations are not known a priori, another approach explored in literature is to utilize on-board sensing [16]. In this work, the authors developed a distributed distance sensor and an optimization strategy for placement of sensor nodes on a robot's body. The distance sensor was integrated into a framework utilizing the *danger field* criterion: a scalar field based on the robot's configuration

and velocity [85]. In the developed framework, the robot attempts to maintain task consistency by hierarchically abandoning tasks according to their specified priority.

Moving beyond modulating robot speed, another popular approach to collision prevention via robot control is through calculation of *potential fields*, first developed in [70]. The approach allows for more complex safety behaviors by defining a field of repulsive vectors that guide the robot's motion, modifying its trajectory in response to dynamically changing factors in the environment. The potential field approach is often deployed as part of integrated safety frameworks that utilize a variety of factors for computation of the field. Examples include combining separation distance, velocity, and an estimate of the affective state of the user [83], incorporating both potential field forces and real, physical forces [51], including the shape and size of the source of danger and relative motion [120], and formulating a risk criterion based on the distance from the robot to the human's head, as well as the human's gaze direction [18].

2.2.2 Post-Collision Methods

Using a variety of control strategies to prevent collisions can be an effective method of improving safety during HRI. However, depending upon myriad factors including the type of robot, sensing system, and assigned task, strict collision prevention is not always possible or practical — in fact, some human-robot collaborative tasks may require a certain level of physical contact. As a result, another body of research has focused on development of control strategies that further ensure safety through detection of and appropriate reaction to human-robot collisions. Figure 2-2 depicts the hierarchy of topics involved in utilizing post-collision control as a safety method for HRI.

The first step toward utilizing post-collision control methods for HRI safety is detecting whether a collision has occurred. As the use of external sensing is often impractical, much of the work in detection and localization of human-robot collisions has focused on methods that incorporate on-board sensing. In [23], for example, the developed system utilizes collision detection and identification signals that can be

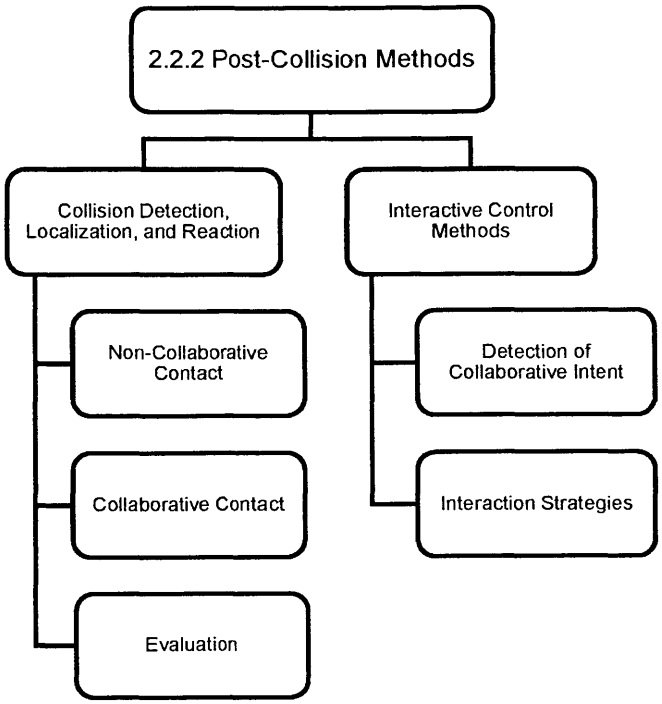


Figure 2-2: Diagram depicting the post-collision safety methods discussed in Section 2.2.2.

calculated with only the joint positions, velocities, and the commanded torques. In a different approach, motor current measurements were used for collision detection and reaction [46]; this technique, in contrast, does not require knowledge of joint velocities or a priroi knowledge of the dynamic model. Another feature of this work is that it is capable of distinguishing between intentional and unintentional contact by monitoring filtered motor currents. This topic was also addressed in [47], where a model of physical contact and insights from observing real impact data were used to form a non-linear Support Vector Machine (SVM) classifier.

Once intentional contact is detected, specialized safety measures and methodologies that differ from the "detection and reaction" paradigm are required. In the framework described in [25], for example, once collaboration was initiated through gesture and speech, contact forces with allowed human contact points (e.g., the hands) were estimated and the robot was controlled such that predefined thresholds were not exceeded, while disallowed contact points (e.g., the head) were avoided altogether. In [38], on the other hand, the authors developed an interactive control scheme for back-

drivable robots in which a human moved a robotic manipulator through continuous contact, and the user's intent was determined through a calculation of control effort based on conservation of momentum.

In addition to developing collision detection and reaction strategies, several researchers have also quantitatively assessed the effectiveness of post-collision control strategies. In work presented in [150], for example, the authors verified the effectiveness of imposing contact force limits by running several physical tests with a real robot and force sensors. In order to test control strategies designed to alleviate contact forces from blunt impacts with even more realism, other researchers made use of instrumented crash-test dummies [52]. Realistic tests of collision reaction strategies were also performed to study soft-tissue injury (such as abrasions, contusions, lacerations, and punctures) caused by robots holding sharp tools [49].

2.3 Motion Planning Methods in HRI

The second main category of methods for improving safety and efficiency in HRI focuses on utilizing specialized motion planners. This involves the development of motion planners that directly consider human presence and movement when computing robot paths and motions, as well as motion planners capable of reasoning on both geometric and task-based constraints and supporting rapid, real-time replanning. These and other related topics covered in this section are depicted in Figure 2-3.

2.3.1 Constraints Based on Human Presence

The primary method by which robot motion planning can serve as a tool for HRI is through direct consideration of constraints related to the presence of human agents, such as distance of separation, human gaze direction, and robot motion legibility. This can be achieved by directly incorporating parameters such as these into the criteria and cost functions used by motion planning algorithms. Doing so enables motion planners to specifically consider how to best plan around the presence of humans,

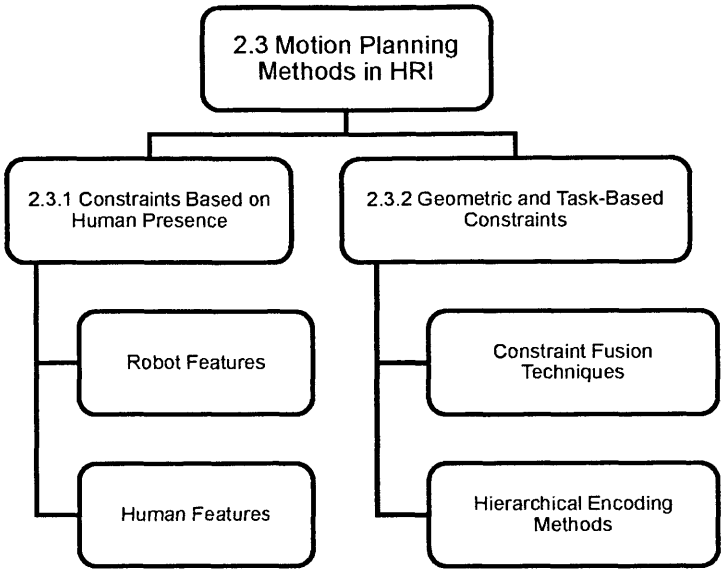


Figure 2-3: Diagram depicting the constraints, features, and techniques discussed in Section 2.3.

as opposed to treating humans as generic obstacles within an environment. Direct consideration of human-based constraints makes these motion planners very useful for improving safety and efficiency in HRI through their ability to proactively avoid motion conflicts and produce comfortable and socially acceptable motions. Furthermore, in contrast with the threshold-based control approaches outlined in Section 2.2.1, parameters can be minimized over the course of motion rather than only when potentially dangerous motions approach safety thresholds. In this way, the relevant safety parameters to be minimized can be kept as low as possible throughout robot motion — and, depending upon how conservative the cost function is, can remain far below predefined safety thresholds.

A variety of planners and frameworks that consider human-based constraints on motion planning have been developed in recent years. In [84], a motion planner applied specifically to robotic manipulation during a hand-off task utilized a danger criterion based on robot inertia and distance from the human to the center of mass of each robot link. Another approach is to make the motion comfortable by limiting jerk and acceleration [135]. This approach has been shown to produce significant differences in terms of comfort as assessed by physiological sensing and question-

naires [26]. Another framework focused on designing not only safe, but also comfortable and socially acceptable robot motions [133]. This was done by considering human kinematics, vision field, posture, and preferences, as well as the legibility of the robot's actions. In [101], similar HRI constraints were considered to drive a cost-based, random-sampling search to plan safe robot motions in cluttered environments. Approaching the problem from a different perspective, researchers looked at the problem of safe motion planning with the human as a passenger of an autonomous vehicle [106]. Toward this goal, the authors developed the Human-Comfortable Path Planner (HCoPP), which takes into account human preferences, such as how far from a wall people like to travel when moving down a corridor, as well as the visibility around corners when approaching a turn.

2.3.2 Geometric and Task-Based Constraints

In order for robot motion planning to be an effective method of improving safety in HRI, the motion planner must be capable of rapid replanning and of taking both geometric and task-based constraints into consideration. The necessity for rapid replanning is due to the inherent uncertainty resulting from the presence of human agents within a workspace. People not only display significant intra- and interpersonal variability and low repeatability of motions [137, 20], but can also unexpectedly change their preferences while performing tasks in terms of which actions are taken or the sequence in which actions are performed. Even the best motion and behavior models cannot account for all human-induced uncertainty, making it important for a robot to have the ability to replan motions quickly should the need arise.

While task models and representations may not be able to capture all potential task variations with perfect accuracy, combining task-based and geometric constraints in robot motion planning for HRI safety has two key benefits. First, while safety regions and buffers can be encoded with geometric constraints, the inclusion of task-based constraints allows for consideration of additional information that could serve to guide a robot, such as where a human is likely to move to or reach for based on prior actions. Second, combining these two constraint types allows for significantly faster

computation through synergistic search-space pruning. In other words, by combining task-based and geometric constraints, it becomes possible to identify configurations that cannot be part of the solution and would not have been identified if task-based and geometric constraints were considered separately.

Several different implementations of planners that utilize geometric and task-based constraints can be found in literature, each combining these constraint classes in different ways and utilizing various specific sub-planners. In [37], for example, the authors introduced a framework that combined high-level, causality-based representations with low-level, geometric reasoning. In this framework, selected geometric models were embedded directly in the high-level representation, which could in turn be modified by the motion planner if no feasible solution exists. The method developed in [119] also used a combination of sampling-based motion planning with symbolic action planning, but integrated these by maintaining a single search tree that was iteratively extended until a dynamically-feasible and collision-free path was found. The motion planning and task planning components interacted through a common value of utility, and actions could be modified by both planners throughout the exploration process. The authors of [19] approached combining task, motion, and manipulation planning through a process of defining and exploring configuration spaces of the robots and other objects in the environment. In the developed method, identification and computation of constraints was done incrementally, with the planner iterating between attempting to find a solution with the current knowledge and searching further in the different configuration spaces under consideration.

Reduced computation time and quicker replanning in combined task and motion planning can also be achieved by utilizing hierarchical encoding schemes. One example of this approach uses vertically integrated hierarchical task networks within a combined action and motion planner and makes use of an algorithm that reuses information about the relevance of state variables to particular subtasks [155]. Another approach is to utilize a structure that allows one to make use of domain-dependent choices when developing the hierarchies, which can result in significant pruning of the search space [66].

2.4 Consideration of Psychological Factors in HRI

The maintenance of *physical safety* often dominates discussion, however ensuring *psychological safety* is also of critical importance, as was mentioned in Chapter 1. It has been experimentally shown that simply preventing collisions right as they are about to occur (i.e., maintaining physical safety) can lead to low perceived safety and comfort [88]. Therefore one cannot ensure safe HRI through maintenance of physical safety alone.

One of the primary methods of ensuring psychological safety during human-robot interaction is appropriate adjustment of robot behavior. Such behavioral adjustments can be split into two categories: those based on robot features and those based on social considerations. Although adapting robot behavior is important for maintaining psychological safety, it is also necessary to evaluate the effectiveness of these adjustments in a principled way. Toward this objective, researchers have developed three tools for assessing psychological safety: questionnaires, physiological metrics, and behavioral metrics. These topics, discussed in the following sections, are depicted in Figure 2-4.

2.4.1 Robot Behavior Adaptation

The first category of research that considers psychological factors of robot behavior involves adjustment of various parameters of the robot's motion, such as speed, acceleration, or proximity to the human, and also investigates how to properly adjust behavior based on the robot's appearance. Although perceived safety and comfort were not the primary objectives of all the works mentioned in the preceding sections, a significant number take psychological safety into consideration by adjusting robot behavior specifically to make interaction more comfortable for humans. This was done by either adjusting robot behavior specifically to make the interaction more comfortable or assessing comfort through questionnaires. Examples of these works include [14], [135], [133], [101], and [106]. While these works took psychological safety into account, there also exists a body of research with the primary focus of evaluating

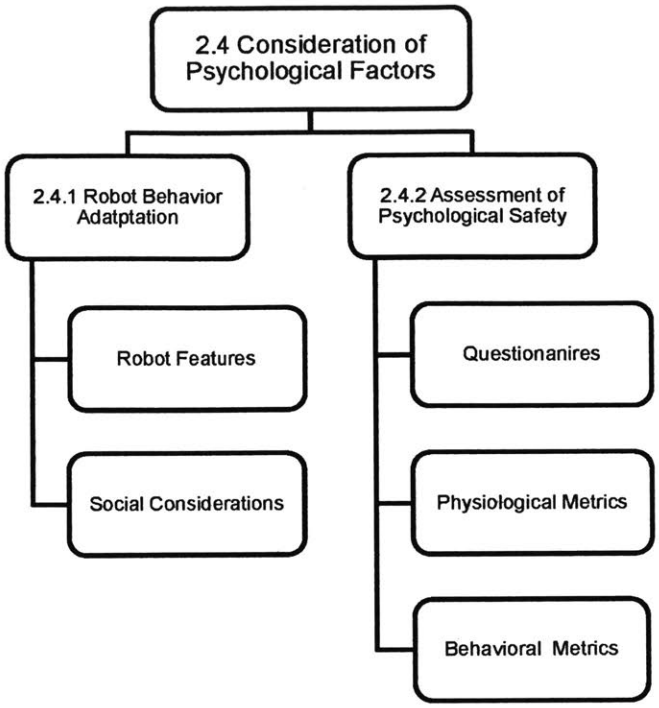


Figure 2-4: Diagram depicting methods of assessing and potential factors influencing psychological safety, as discussed in Section 2.4.

how robot behaviors affect psychological safety factors, both in terms of studying and adjusting robot features, as well as understanding the impact of social factors.

Researchers across various fields of study have shared the concerns and sentiments regarding psychological safety expressed in the above works, leading to the study of the possible negative psychological effects of numerous aspects of robot behavior. It has been shown, for example, that robot speed, separation distance, and advance notice of motion affect human stress as assessed via physiological sensing and questionnaires [9]. Similar features have also been studied in unconstrained scenarios without predefined interactions using real-time reporting of discomfort via a hand-held device [75]. In order to obtain a more objective measure of real-time psychological state, another approach is to use physiological sensing (skin conductance response, heart rate, and corrugator muscle activity) and a fuzzy inference engine [82]. In this work, the authors compared participants' response to a co-located robot with different speeds and motion types. In addition to evaluating the effects of robot speed and distance, researchers also studied the effects of robot body design on participants' levels of per-

ceived comfort during execution of several robot behaviors [17] as well as the effect of acoustic communication using beeps of rising or falling intonation [41].

The second main thread of work in psychological safety focuses on studying the impact of a robot's social behavior during human-robot interaction. This involves discovering which social behaviors observed during human-human interaction are important for HRI, how failing to follow social conventions during HRI impacts psychological safety, and how personality traits, culture, and demographic factors can affect these considerations. In one example of this work, researchers investigated participants' response to a robot invading their personal space and studied people's expectations of a robot following social norms in this regard [64]. The authors found that the participants' reaction to a robot invading their personal space were stronger than when compared to a human doing the same. In addition to proxemic distance, researchers also investigated how "power distance" (the role of the supervisor vs. the subordinate) and "task distance" (cooperative vs. competitive performance) can affect user experience and comfort during HRI [71]. Other work has shown that people's personalities and experiences, such as prior experience with robots or pet ownership, can also impact their comfort when interacting with a robot [142]. In addition to studying these qualities, researchers were also able to show that the robot's "personality," including likability and gaze behavior, affected physical and psychological distancing (i.e., how much information the human discloses to the robot) [108].

While proxemic preferences are a function of personal characteristics, these preferences can shift over time and can even vary from culture to culture. For example, the results of a 6-week study designed to assess participants' proxemic preferences when interacting with robots in homelike environments indicate that the majority of adaptation of preferred interaction distances occurred within the first few sessions, then stabilized afterwards [154]. The seminal work presented in [53] indicated that people of different cultures have significantly different standards for maintaining personal space, and it has been empirically demonstrated that such standards also manifest for HRI [65].

As proxemic preferences and other social standards have been found to have an impact on psychological safety, it is important for robots to be able to reason about these conventions and adjust their behavior appropriately. Frameworks such as the one developed in [72] can be quite useful for this purpose: in this work, social conventions are treated as constraints in a planning problem, which is solved to select socially acceptable paths for the robot. Another method for incorporating social norms is to utilize human demonstrations to build a model of appropriate behavior, which can then be incorporated into the robot's motion planner [39].

2.4.2 Assessment of Psychological Safety

In order to determine whether certain robot behaviors negatively affect psychological safety, and to ensure that the proposed behavior adaptations are capable of mitigating these negative effects, it is necessary to assess psychological safety in a principled manner. Assessments of psychological safety can be split into three main categories: questionnaires, physiological metrics, and behavioral metrics. These three distinct methods of assessment differ in key qualities, such as whether or not they can be collected in real-time, whether they are subject to interpretation or purely objective, and how intrusive or disruptive the measurement can be to the human. These techniques represent valuable tools for researchers to ensure that psychological safety is properly taken into consideration in HRI.

Several validated questionnaires developed in prior work have proven useful for the assessment of psychological safety. One example of such questionnaires is "Godspeed", a carefully designed set of questions based on prior work in HRI that allows researchers to measure several key HRI metrics on a standardized, validated scale, including perceived safety [11]. Another example is the Negative Attitude toward Robots Scale (NARS), which is composed of three sub-scales assessing "situations of interaction with robots, " "the social influence of robots, " and "emotions in interaction with robots" [113]. More recently, researchers developed BEHAVE-II, a method that utilizes both subjective and objective metrics to assess human responses to robot behavior by studying two types of responses: attitudinal and behavioral [64].

In addition to developing and validating new questionnaires and metrics specifically designed to assess perceived safety and comfort, researchers have also incorporated such questionnaires into their experiments. Several works mentioned in the preceding sections included questions about perceived safety and comfort within their assessments, even if this was not the main topic of the research [135], [26], [106], [138], [17], [71].

While questionnaires can be a useful method of assessing psychological safety, this method also has certain drawbacks. Questionnaire items are generally open to interpretation, which can lead to unexpected biases and noise among responses. Furthermore, questionnaires can be impractical if perceived safety is to be tracked outside of controlled experiments, either for online behavior adjustment or logging for future analysis. As a result, researchers have also considered analysis of physiological metrics as a potential method of assessing psychological safety during HRI. In one example of such work, a variety of physiological signals were studied to determine which ones could be utilized in estimating human affect during HRI [82]. The authors found that a fuzzy inference engine based on five physiological features, namely heart rate, heart rate acceleration, skin conductance response, rate of change of skin conductance, and corrugator muscle response, is effective at estimating affect. It has also been shown that robot speed, advance notice of motion, and separation distance all influence the skin potential response signal [9]. Furthermore, the authors showed that questionnaire responses did not always reflect stress detected using physiological sensing, suggesting that some stresses encountered during HRI may not be consciously perceived by the human.

Another method of assessing psychological safety is through observation of human behavior in response to a robot. Similarly to physiological metrics, this method does not rely upon self-reported information, which is susceptible to variations in interpretation. One advantage of utilizing behavioral metrics over physiological signals is that physiological signals are often difficult to analyze, as a variety of emotions can affect them. (Skin conductance, for example, can increase when a person feels frightened or happy [8, p. 177].) Carefully selected behavioral metrics can be a more direct proxy

for perceived safety and comfort. For example, one study evaluated human response to being approached by a robot according to, among other metrics, average and minimum distance the person maintained from the robot during interaction [142]. In another study, the authors not only utilized the distance the participant kept to the robot, but also how much information the participant was willing to disclose to the robot [108]. Another interesting behavioral metric used in literature is to not only track how closely people prefer to approach the robot themselves, but also have the robot approach participants and give them the ability to move the robot forward or backward to their preferred distance of separation [154].

2.5 Prediction Methods in HRI

The final main category of methods utilized for improving safety and efficiency in human-robot interaction are those based on prediction. In some HRI situations, it is reasonable to use motion planning methods (such as those described in Section 2.3) to improve safety and efficiency of the interaction by assuming that the environment is quasi-static and simply rely upon replanning motions quickly when the movement of human and robotic agents conflicts with the initial plan. However, this approach is not appropriate within more dynamic environments. In this context, motion plans based on a quasi-static assumption quickly become obsolete, making reliance on replanning impractical — particularly if humans and robots are working in close proximity to one another, as there may not be sufficient time to replan. As a result, the ability to anticipate the actions and movements of members of a human-robot team is critically important for providing safety and efficiency within dynamic HRI environments. First, I will briefly discuss prior work on making the robot predictable to the human with the use of explicit and implicit cues. Next, I will describe methods of predicting human actions based on sensor input and action labels. Finally, I will discuss prior work on human motion prediction, which is the focus of this thesis.

2.5.1 Prediction of Robot Motions and Actions

Human-robot collaboration is a two-way interaction; as such, human agents' ability to predict the actions and movements of a robot is as essential as the ability of a robot to predict the behavior of humans. By taking the predictability of robotic teammates into account, researchers can develop both implicit and explicit methods of conveying robot intent, allowing human teammates to accurately predict the behavior of their robotic coworkers. These topics are summarized in Figure 2-5.

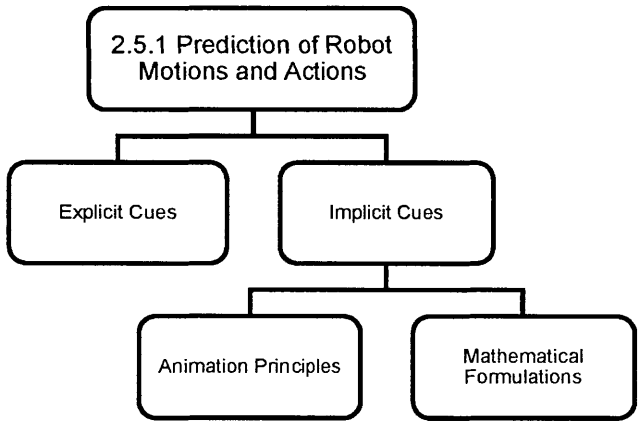


Figure 2-5: Diagram depicting the types of cues that can aid in prediction of robot motion, as discussed in Section 2.5.1.

One method of making a robot more predictable to human teammates is via explicit communication of intent, where the robot directly communicates it's planned actions and motions to the human. In [136], for example, verbal feedback from the robot is utilized when coordinating a joint activity between a human and robot, leading to faster task execution and more positive ratings of the robot as a teammate. In addition to speech, researchers also studied the effects of communicating joint plans via the robot's gaze [87]. Another approach investigated how to communicate the motion of flying robots to humans in a shared workspace. The study specifically looked at the speed and accuracy of human prediction of the free-flying robot's motion based on various LED array communication patterns [139].

While direct, explicit communication can be an effective method of conveying robot intent, implicit cues can also communicate future robot motions and actions.

Using our understanding of human-human teaming as a basis, the benefits of implicit communication of intent are readily apparent, as one would not expect a coworker to always explicitly communicate where they are about to reach or move toward. Instead, people convey intent through subtle cues embedded in the ways they perform their motions. Applying this concept to HRI, the authors of [141] investigated the use of animation principles to aid in the interpretability of robot intent. This involved, for example, having the robot change its height to acknowledge the human, or directing its gaze at the human and then at an object about to be manipulated. While these animation principles improved the interpretation of activities upon completion, animation principles were also shown to improve predictability of robot movement [138]. In particular, this work focused on implicit communication of motion intent for assistive free-flying robots through animation-inspired modification of motion primitives. Research in [33] also investigated how robot motion could implicitly convey intent, but approached this problem through the development of mathematical definitions of predictability and legibility of robot motion. The paper discussed how *legible* (intent-expressive) and *predictable* (matching expectation) motions can be quite different, and that these properties of motion are often contradictory. For example, in order to be legible, the motion often has to be more exaggerated, making it less similar to what a person might expect the motion to look like. In later work, the effects of utilizing legible and predictable motion on the quality of human-robot collaboration were analyzed [32]. In the developed study, participants performed a collaborative task with a robot in which they needed to infer the robot's intent to complete the task efficiently. By implicitly conveying the robot's intent and making it more easy to predict, people working with the robot can more effectively plan their own motions such that they remain safe.

2.5.2 Human Activity Prediction

By predicting which action a person might take next, robot motion and activity planners such as those described in Section 2.3 can identify actions and paths that will result in safe and efficient interaction. This selection process can involve adjusting

the action the robot will take, the timing of when that action should be performed, or the motions the robot will perform in order to achieve the action. Work in this field has incorporated a wide variety of methods, with researchers often formulating probabilistic models and frameworks according to which prediction could be made. Among these approaches, some reason directly on low-level features derived from cameras, depth sensors, and motion capture systems, while others reason on abstract representations of actions or task steps in order to predict future activities. Beyond predicting the next human action, other work in this field focuses on predicting the timing of the actions, which allows a robot to not only decide what action to take in order to maintain safety, but also when it would be best to take that action. These topics are summarized in Figure 2-6.

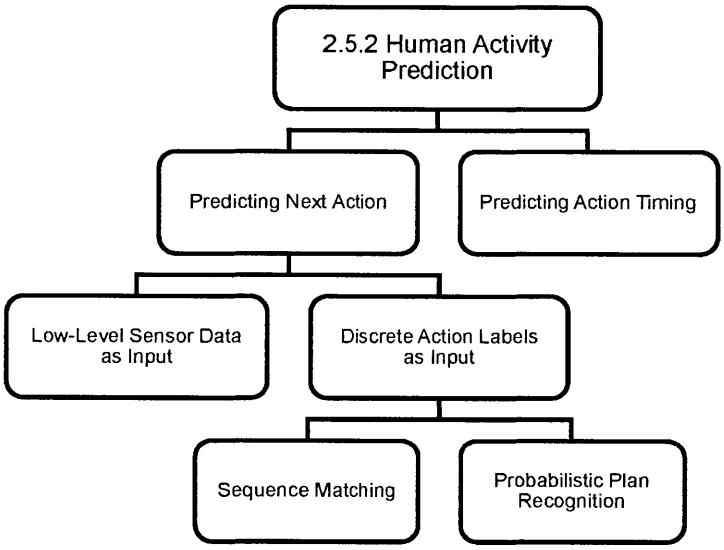


Figure 2-6: Diagram depicting the methods for human activity prediction discussed in Section 2.5.2.

Work in the field of activity prediction that utilizes low-level representations of sensor data as input has included a wide spectrum of prediction techniques, ranging from detection of activity processes (such as walking or pouring a liquid) to action goals (such as what item a person is about to grasp). With regard to activity processes, researchers used various approaches, including using integral histograms of spatio-temporal features derived from RGB videos [126], a conditional random field (CRF) augmented with anticipated temporal segments that made use of RGB-D input

and object affordances [79], and first person RGB video to detect subtle sub-activities called *onset activities* that occur just prior to the main activity [127].

While the ability to predict activity processes can provide task-level information to task and motion planners, in the context of ensuring safety in HRI, the ability to predict action goals (such as where a person might reach toward) is also very important. This is especially true during close-proximity collaboration, where simply slowing down and stopping the robot via control-based methods (as discussed in Section 2.2) can lead to constant motion conflict and many stressful near-collision situations. The authors of [102], for example, developed a framework that utilizes labeled demonstrations of reaching motions to develop models for prediction of workspace occupancy. In this framework, separate Gaussian Mixture Models (GMMs) were trained for each reaching goal position of a particular task and Gaussian Mixture Regressions (GMRs) were used to generate representative reaching motions. Work in [117] also focused on predicting reaching locations based on human demonstrations, but with a time-series analysis that utilized multivariate Gaussian distributions over the tracked degrees of freedom of the human arm defined for each time step of the motion. The paper also explored how task-level prediction, incorporated into the framework as a prior in the Bayesian formulation, could affect prediction results as a function of the task-level prediction's accuracy. Another more recent approach focused on not only detecting and predicting activities, but doing so in a human interpretable way [57]. This was accomplished by formulating classifier ensembles from a collection of independent multivariate GMMs that model the motions of individual objects (such as human joints) during task execution. In this manner, the approach can provide a description of what specific objects were closely matched to a given activity and which ones were not. Furthermore, the developed method is also resilient to time variance due to the use of a modified form of max-pooling.

Rather than reasoning on low-level sensor data directly, a second class of human action prediction techniques utilizes task models to reason on what actions have been taken, and then uses these data to inform prediction. In these works, the purpose of the sensing platform is action detection. A framework developed in [60], for example,

utilized a cost-based Markov process to anticipate human actions and select robot actions based on confidence of the validity of the prediction and risk. In [111], the collaborative task was encoded as a Markov Decision Process (MDP), and it was shown that observing changes to the entropy rate of the derived Markov Chain could be utilized to encode the uncertainty of the robot about what action the human will perform next. An extension of the MDP called the Mixed Observability Markov Decision Process (MOMDP) was later utilized in [109]. In this work, user models are learned automatically from joint-action demonstrations by clustering action sequences and then learning reward functions for each cluster through inverse reinforcement learning. While a majority of the previously mentioned works focus on short-term prediction of brief actions, the authors of [94] developed a framework for predicting actions of longer duration composed of constituent actions with complex temporal compositions.

The timing of an action is another important aspect of human action anticipation. In one example of work on this topic, researchers utilized a probabilistic graphical model in which the start and end time of actions are treated as probabilistic variables conditioned on earlier events and considers the uncertainty in sensing executed actions and variability in timing among individual humans when making its predictions [56]. The authors later extended this work to consider not only uncertainty in timing and action detection, but also more complex task descriptions containing ambiguous task ordering [55]. In more recent work, the authors of [160] developed an approach for predicting both task sequences and action timings with the use of higher-order Markov Chains. In this method, which is capable of learning online during execution, the probabilities of future action chains are predicted as a function of previously executed tasks. Then, using these predicted sequences of actions and previously observed action durations, the method provides a distribution over timings of human actions that require the robot's assistance. The results indicate that using this technique, as opposed to a purely reactive robot, leads to better task performance. This has also been demonstrated in the context of human-robot handovers [62]. In addition

to improving task performance, this study showed that utilizing prediction of action timing can lead to a better user experience as well.

2.5.3 Human Motion Prediction

Although the ability to predict human actions can be useful for generating safe robot motions and action plans, knowledge about what activity will be performed or the end location a person is reaching or walking toward does not provide information about which specific portion of a shared human-robot workspace the human will occupy during the execution of that predicted action. This additional information could be leveraged to ensure safe robot motion by enabling the robot to reason not only on the expected start and end locations, but on the entire expected human motion.

The basis of human motion prediction can be divided into two distinct categories: goal intent and motion characteristics. For the former, action prediction can often serve to inform motion-level prediction by inferring humans' goals, which, in combination with an appropriate motion model, can be used to anticipate how a human will move as he or she walks or reaches toward the predicted goal. In the latter category, motion prediction is not linked to predicted goals or actions, but instead utilizes techniques such as analysis of biomechanical predictors or reasoning on features of natural motion. These methodologies for human motion prediction are summarized in Figure 2-7.

Goal Intent

When predicting human motion based on goal intent, the system infers a human's goal and predicts the path or trajectory he or she is likely to take in order to reach that goal. This can include both smaller manipulation movements and larger movements involving ambulatory motion.

As goal prediction is necessary for this approach, several of the action prediction methods described in Section 2.5.2 could be used for this purpose. Some of the approaches mentioned earlier also predict the motion people could take while performing

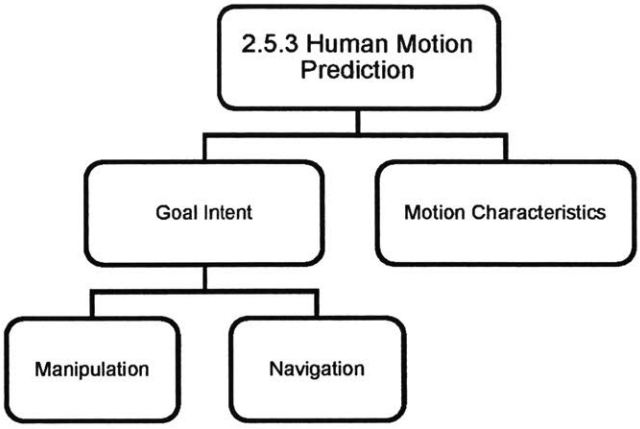


Figure 2-7: Diagram depicting the methods for human motion prediction discussed in Section 2.5.3.

the predicted manipulation actions. In the work presented in [102], the system uses regressed motions derived via the GMR to compute swept volumes that define human workspace occupancy during execution of the predicted reach. This approach was later extended to build the motion models in an unsupervised fashion (i.e., without requiring a labeled training set) [98]. The developed method utilized an iterative approach where each successive trajectory is compared to the current GMMs and then is either incorporated into one of these models via iterative expectation maximization, or is used to initialize a new GMM. The algorithm decides which of these options is taken based on whether the likelihood of the trajectory being generated by any of the current GMMs surpasses a specified threshold. In order to improve the accuracy of prediction, the approach uses a two-layer framework where separate GMMs are used for motions of the arm and palm.

In the previously discussed work described in [117], on the other hand, prediction during a reaching motion is calculated using multivariate Gaussian representations. In contrast to computing a swept volume of the entire reaching motion, the models computed in this work describe the mean and variance in the human's hand position for each of the possible goal positions at each time step, yielding a prediction of human position as a function of time during the reach.

Prior work has also treated human reaching motions as Markovian, training Hidden Markov Models (HMMs) in order to predict future motion [27]. In this work,

the authors developed a technique that first uses example trajectories to construct a topological map of the motion using the Instantaneous Topological Map algorithm. The nodes of this topological map then become discrete states to be used in the HMM structure. Once the set of states is updated, the next step is to use the new trajectory to update the model parameters, such as initial state probabilities and the state transition matrix. In order to predict future motion, the HMM is then used to compute the conditional probability, given the partially observed motion, of occupying any portion of the space at a future time horizon of interest. This allows for the construction of a time-varying, probabilistic safety region.

In another example of work mentioned in the previous section that also extends to motion [79], once the system predicts an action, it utilizes Bézier curves to define potential trajectories of the human hand while performing the action. An extension of this work allows for more-detailed prediction of human motion during action execution [63]. In this work, a low-dimensional representation of a high-dimensional model of human motion is computed through a Gaussian process. The low-dimensional description of the motion is then incorporated as latent nodes in a CRF representation to form a model called a Gaussian process latent conditional random field. By learning a two-way mapping between the high- and low-level representations, this approach allows for computationally tractable prediction of high-dimensional motion while maintaining the ability to reason on relationships between people and objects in the given scene. In addition to being able to predict human motions with higher fidelity, the compact representation of motion allows for reasoning on the physical plausibility of actions, thus improving prediction.

Another manipulation motion prediction technique was specifically designed for reaching motions performed during known, collaborative tasks based on inverse optimal control [103]. In this work, example data of two people performing a co-manipulation task were collected via a motion capture system. The gathered trajectories, along with feature functions encoding smoothness and distance relationships, were then used as inputs for the path integral inverse reinforcement learning (PIIRL) algorithm in order to produce the reaching motion cost functions. The approach uses

these cost functions and a human kinematic model with 23 degrees of freedom to predict the human's motion by iteratively replanning the motion with the stochastic trajectory optimization for motion planning (STOMP) algorithm. The authors found that the PIIRL algorithm is capable of correctly recovering the cost functions for sampled motions created by manually setting weights. Furthermore, the developed framework can predict human motion more effectively than hand-tuned cost functions.

Prior work has also investigated integrating action prediction with short-term motion prediction for intent-aware motion planning [115]. In this work, the joint human-robot task is modeled as a Markov Decision Process, and the robot's policy is learned via Q learning. The next human action is predicted based on previously observed sequences of actions, to allow the robot to select an appropriate action based on the learned policy. The current human action is estimated using an Import Vector Machine (IVM) which is trained using example trajectories and uses joint velocities, accelerations, and positions, as well as the current progress of the task as the feature vector. Short term motions (on the order of one second) are predicted by modeling the motion as a Sparse Pseudo-input Gaussian Process. Independent models are trained based on each discrete progress state of the joint task. An online motion planner then takes the predicted short term motion as an input for collision avoidance.

Goal-based human motion prediction can also be applied to larger movements involving ambulatory motion, such as walking. In one example of such work, the authors developed a two-step approach incorporating growing hidden Markov models (GHMMs) and the social forces method [34]. During the learning step of this approach, the system uses GHMMs to learn typical human walking patterns from collected data, allowing for continuous updating of the model as new data are collected. In the second step, the system combines goals inferred from partial trajectories using the learned GHMMs with a motion model based on the concept of social forces from work in [97] in order to predict the path a human will take toward their goal based on static obstacles, other people, and the physical constraints of the environment.

An earlier approach uses no explicit motion models; instead, learning is performed automatically by clustering human motions via the expectation maximization (EM) algorithm and modeling these clustered motions using hidden Markov models (HMMs) by [12]. In this approach, natural *resting places*, or places where the human's motion slows or pauses, are identified from the data and used as potential goal locations.

The authors of [163], on the other hand, developed a goal-based human walking trajectory prediction method that leverages the assumption that people move efficiently when navigating a space by modeling human motions using maximum entropy inverse optimal control. One key benefit of this approach is that the cost function it learns is a linear combination of features based on environmental obstacles, allowing it to generalize well in the event that objects within the environment are moved to new locations or removed from the scene altogether. Furthermore, the presented approach uses an incremental planner that enables real-time deployment by developing cost maps without taking prediction of motion into account, and then iteratively plans a robot trajectory while simulating human motion.

The influence of environmental factors was also considered in the work presented in [67]. In this work, which is specifically focused on long-term prediction of pedestrian motion, the authors model pedestrian motion as a discrete-time jump-Markov process with their goal as a hidden variable. To incorporate environmental factors, the reward function of the MDP is a function of a *semantic map* that describes different regions such as roads or sidewalks. Furthermore, the approach also accommodates time-varying environmental factors, such as traffic lights, by modifying the reward function as a function of world state.

It is also important to consider the confidence of the predictor when using it to plan safe and efficient robot motion. Toward this goal, recent work developed a method of characterizing the current prediction confidence based on the extent to which the motion of the human matches the model and incorporating this confidence metric in a Bayesian framework [40]. The authors then incorporated the resulting

probabilistic prediction with a real-time motion planner to plan probabilistically safe, confidence-aware paths.

Motion Characteristics

Some human motion prediction methods do not rely upon estimates of goal locations, but make use of observations about how people move and plan natural paths. This class of techniques encompasses a variety of approaches, including discovery of likely motion progressions, use of biomechanical predictors, consideration of features of natural motion, and general unsupervised approaches for learning about how agents move within an environment.

In one example of such work, the authors of [140] use motion capture data to encode skeletal motion patterns as HMMs, which are then grouped via Ward clustering to form a hierarchical structure called the "motion symbol tree." The system then learns sequences of motion symbols through the use of Ngrams, forming a directed graph — the "motion symbol graph" —that represents transitions between motion patterns, and thus causality among human behaviors. The motion symbol graph is then used in conjunction with current motion observations to predict future motion patterns, represented as skeletal motions.

Another approach is to first decompose data into high-level classes in order to develop more specific models for different types of human activities [158]. In this work, the authors used previously observed human trajectories to train an SVM classifier that first decomposes the data into classes like *wandering* or *stopping*. The framework then forms clusters within these classes using the partitioning around medoids (PAM) algorithm, with a modified distance function that allows for better clustering of similar, non-overlapping trajectories, especially with limited movement. These clusters are then used to extract prototypes, which are matched to observed partial trajectories to enable prediction. As the clustering is performed in an unsupervised fashion, one key benefit of this method is the ability to utilize new trajectories to further refine and adapt prediction.

Prior work has also shown that utilizing lower-level, biomechanical turn indicators can also inform motion prediction [145]. In this work, the authors used a motion capture system to collect data of people walking toward targets in a room. Results from statistical analysis showed that indicators such as head orientation and body velocity normalized by height can signal a human's intention to turn prior to physical performance of the turn itself. This allows for prediction of turning motions without necessarily predicting a goal location. The authors also applied these turn indicators within a goal-based prediction framework based on the previously mentioned work described in [117] to show that these indicators provide a signal strong enough to be used for motion prediction. It is important to note that the biomechanical turn indicators validated in this study can be incorporated into other prediction frameworks as well. The final step taken by the authors was to test the utility of the prediction in a dynamic environment via a closed-loop simulation in which a robot planned its motions according to predictions about human motion. They found that by reasoning on the predicted human motion, the robot was able to take paths that avoided motion conflicts; the resulting robot path plan would allow for safer and more comfortable co-navigation of the space.

Several other works have also investigated biomechanical turn indicators, considering parameters such as gaze yaw, head yaw, trunk yaw, trunk roll, trajectory heading, and body center of mass medio-lateral displacement [95]. In their review, the authors of [95] found that head yaw was the most reliable kinematic variable, followed by trunk roll.

In order to make longer term motion predictions without first specifying a goal, another technique used in literature is to model motions as velocity fields with the use of Gaussian Processes (GP). Recent work, however, has shown that using GP likelihood as a similarity metric during unsupervised clustering of motion patterns can produce poor results (e.g., splitting trajectories with small planar shifts into separate clusters) [22]. To address this, the authors created a new Bayesian non-parametric method called the Dirichlet process active region (DPAR), which discretizes the environment into a grid and uses a max-pooling step to avoid separating trajectories with

small planar shifts into different clusters. The approach led to a significant improvements in prediction accuracy and computational efficiency as compared to using the Direchlet Process Gaussian Process (DPGP).

While the above works consider the motion of the human independently of other agents in the environment, one interesting and useful insight is that the trajectory of the robot affects that of the human. Based on this concept, researchers used demonstrations of human motion to develop a method of motion prediction that learns joint trajectories [81]. By analyzing trajectories during human-human interaction, the researchers investigated which features of walking motion the system could use to learn how to characterize and predict typical human walking behavior. The model, based on the principle of maximum entropy, considers features including the amount of time needed to reach a goal, acceleration profiles, walking velocity, and collision avoidance behavior. This approach, based on the physical aspects of the trajectory (as opposed to a Markov model), considers how people reason over these features when planning joint trajectories. (Humans might, for example, prefer to maintain a steady walking speed, try to reach their goals quickly, or attempt to minimize proximity to obstacles and other agents.) The developed method was validated both in simulation and in a physical environment, indicating that human trajectory prediction based on consideration of joint trajectories can successfully guide a robot in a natural and socially compliant way.

The influence of agents on each other's motion was also considered in a more recent work that, unlike the research presented above, does not rely on hand-crafted features, but instead makes use of deep reinforcement learning [21]. The approach learns a value network that, given the current positions and velocities of the agent and its neighbors, estimates the agent's time to goal. Training of this value network occurs in two phases. First, the network is initialized by generating training trajectories with another collision avoidance algorithm. Next, the value function is refined via reinforcement learning by simulating scenarios with two agents and running an ϵ-greedy policy (i.e., having the agent selecting a random action with probability ϵ and following the candidate value network otherwise). Once the value network is trained,

paths are generated by maximizing a one-step lookahead value (i.e., one that leads to joint state with highest value) based on the current joint configuration.

Besides collision avoidance and spacing constraints, there are other interesting social considerations that can be exploited for human motion prediction in the context of multi-agent interactions. The authors of [124], for example, considered the effects of social grouping constraints in the context of long-term prediction. The approach models the movement of agents as a Markov Decision Process and uses a random walk algorithm to project likely paths forward based on a learned stochastic policy. The results of this work demonstrate that combining long term prediction via the MDP planner with local social forces based on social grouping improved prediction accuracy on a real human motion dataset.

Interaction with other agents in the environment is not the only context of a scene that can be utilized for prediction, however. Within the computer vision community, researchers developed a method that utilizes an unsupervised, data-driven approach for visual prediction using a static image [152]. In this work, the system learns relationships between subsections of the image to form a context-based Markov model, such that no prior assumptions about agents and activities are necessary. This work provides a visual prediction of the future appearance of a scene by projecting the position and appearance of subsections in that scene. While this approach was not specifically designed for human motion prediction, the unsupervised, context-based method developed here could easily be applied to this domain, as human movement within an environment is also influenced by the context of the scene.

2.6 Utilizing Human Motion Prediction for Safety and Efficiency in HRI

In the preceding sections, several distinct types of methods of providing for safe and efficient human-robot interaction were introduced. Each of these types of methods has unique benefits and drawbacks that have important implications on the quality

of the interaction. In light of the presented methods, the goal of this section is to motivate the use of human motion prediction as a particularly suitable technique in a variety of HRI contexts. Next, I will expand upon the motivations described in Chapter 1, and describe the main roadblocks to deploying human motion prediction in practice that this thesis aims to address.

2.6.1 Key Advantages of Human Motion Prediction

In the preceding sections four main types of methods for safe and efficient HRI were introduced: control, motion planning, psychological factors and prediction. In terms of improving safety in HRI, many of the works presented in Section 2.2 provide robust and reliable approaches. As previously discussed, post-collision methods can be effective in detecting collisions and minimizing harm through appropriate response. While these serve as an important last effort in minimizing harm, the more proactive approach of pre-collision control methods provides another layer of safety. From this category, techniques based on quantitative limits of parameters such as velocities, potential impact forces, and energy allow systems to provide quantitative guarantees about the robot's ability to inflict harm in the event of unwanted collision. One of the main benefits of this approach is that these guarantees are provided without having to rely upon accurate and robust detection and tracking of co-located humans. In many contexts, however, the human might only be in close proximity to the robot sporadically, making these pre-collision methods potentially overly conservative and inefficient. The techniques based on speed and separation monitoring allow for greater flexibility than strictly limiting the robot using parameters such as energy or velocity, but they also require a low-latency implementation with robust tracking of the human within the given space. Techniques based on the potential field approach, on the other hand, allow for implementation of more complex collision-avoidance behaviors beyond simply adjusting the robot's speed, but their efficacy is directly linked to the strategy used to construct the potential field.

While these more complex pre-collision methods can be effective in preventing unwanted collisions, results from an experiment assessing close-proximity human-

robot collaboration indicate that simply preventing collisions as they are about to occur can lead to inefficient human-robot interaction and negatively impact perceived safety and comfort [88]. In this experiment, participants performed a collaborative task with a robot operating in two distinct modes: a standard mode in which the robot determined the quickest path to its goals and employed a pre-collision safety system based on separation distance to slow and stop its motion [90], and an adaptive mode in which the robot used human-aware motion planning to avoid portions of the shared workspace that it expected the human to occupy. The results of the experiment showed that utilizing the adaptive mode led to better team fluency, as well as increased satisfaction with the robot as a teammate and higher perceived safety and comfort.

The fact that, in certain scenarios, collision prevention and alleviation through low-level control has been shown to lead to significantly poorer safety and efficiency compared with human-aware motion planning provides significant motivation for utilizing motion planning as a method of improving the quality of HRI. As described in Section 2.3, this class of methods can serve as an effective tool by encoding both physical and psychological safety into the motion planners' cost functions. This technique allows for a more proactive approach to ensuring safety and efficiency compared with the control-based methods discussed in Section 2.2. Importantly, motion planners enhanced with consideration for human-robot interaction can be used for both manipulation and navigation planning and can be applied to virtually any robotic platform, indicating the versatility of this method. Due to the dynamic nature of any environment occupied by people, however, such planners must be able to rapidly recompute new paths and motions. As described in Section 2.3, this is often achieved by combining task and motion planning to aid in efficient traversal of a complex search space, and by utilizing hierarchical encodings of the constraints.

There are practical limits, however, to replanning based solely on the current configuration of a rapidly changing world state. Consequently, while in some situations it is reasonable to use motion planning methods to improve safety and efficiency of the interaction by assuming that the environment is quasi-static and simply rely upon replanning motions quickly when the movement of human and robotic agents conflicts

with the initial plan, this approach is not appropriate within more dynamic environments. In this context, motion plans based on a quasi-static assumption quickly become obsolete, making reliance on replanning impractical — particularly if humans and robots are working in close proximity to one another, as there may not be sufficient time to replan.

Consequently, the ability to anticipate the actions and movements of members of a human-robot team is critically important for providing safety and efficiency within dynamic HRI environments, which motivates the use of prediction-based methods discussed in Section 2.5. Predicting the actions a person will take allows the robot to select its own actions in a way that supports safety and efficiency. For example, the robot can avoid taking actions near the location of the expected human action or plan an action sequence that will ensure it will time its assistance in an efficient way.

While these action-level robot adaptations can certainly be effective in some scenarios, their effectiveness can be limited when motion conflicts can occur outside of action regions (i.e., en route to the location where the action is preformed). In these situations, reasoning only on what actions will be taken by the human might not be sufficient to support safe and efficient interaction, as a motion conflict can occur anywhere in the region shared by the human and robot. Furthermore, if the robot's next action is not flexible, only knowing what action a co-located human will take does not necessarily give the robot the ability to efficiently and safely navigate the shared region.

The key to supporting safe and efficient interaction in these types of scenarios is utilizing the human motion prediction methods described in Section 2.5.3. By building upon action prediction and predicting not only what goal the human will go toward but also what path they will take toward this goal, a robot gains the ability to make adaptations on the motion planning level in order to avoid potential motion conflicts. When combined with planning in time, this allows for sophisticated adaptive behaviors in which the robot can intelligently move through a shared region in a safe and efficient way. This topic will be discussed in Chapter 5, where this claim

is confirmed through implementation of human motion prediction with planning in time on a real system as well as in simulation.

In summary, in context of all of the aforementioned methods of providing for safe and efficient human-robot interaction and their limitations, human motion prediction is particularly effective in a wide variety of human-robot interaction scenarios. Due to the rich information gained from predicting human motion, this method is particularly versatile and powerful. By building upon the other methods, this technique allows for proactive behavior that is not possible with control, motion planning, or action prediction based techniques alone.

2.6.2 Roadblocks to Deploying Human Motion Prediction

As introduced in Section 1.1, While human motion prediction can be an effective tool for safe and efficient human-robot interaction, deploying this method in practice can prove to be challenging. Due to the wide variety of contexts in which HRI can exist, without technical expertise in prediction, it can be difficult to determine what particular method of human motion prediction should be used for a given task and environment. As many of the methods are implemented with specific types of tasks or motions in mind, selecting an inappropriate method can result in poor prediction performance. Furthermore, as introduced in Section 1.1, goal-based methods that rely on matching partial trajectories to a full, representative trajectory can often provide inaccurate predictions due to poor trajectory alignment, greatly limiting robustness and usability.

Difficulty in Predictor Selection

One of the main difficulties with deploying human motion prediction outside of a laboratory setting is the difficulty of selecting an appropriate prediction method for a given human-robot interaction scenario. Many key questions need to be asked in making such a selection, and the answers are not necessarily straightforward. One such question, for example, is whether or not there is a finite set of distinct possible

actions the person can take during the interaction. If yes, then goal-based approaches like the ones introduced in [117], [102], [98] or [57] that predict which of these actions is being taken could be the best choice. It's possible, however, that the time series describing these individual actions are not distinct enough for the underlying classifiers to perform well, and this negative impact can vary from method to method. If none of the action classification methods perform well, then perhaps it could be useful to leverage action sequences using method such as those described in [115], [160], or [109].

If the given scenario does not contain a finite set of goals, then methods based on motion characteristics could be more useful. If there are distinct types of movements, then approaches like the one introduced in [158] could be used. If not, then perhaps using biomechanical indicators like in [145] could provide useful predictions. As discussed in [95], however, there is a large variety of studied biomechanical indicators, and they produce different, and often contradictory, results depending on the given scenario, making selecting the best biomechanical indicator for a given task difficult. Considering the impact of the motion of the robot or other people could also be useful, where approaches like those introduced in [22] or [81] can be used. Furthermore, if the prediction involves groups of people then the method described in [124] could be a good choice. It's possible, however, that the environment in which these methods are being deployed in impacts their efficacy. Consider, for example, how these types of social motion adaptations differ in a factory and in a shopping mall. In a factory, people might be focusing more on getting somewhere quickly without necessarily keeping the same spacing or using the same evasive movement as a group of people walking in a mall.

On a different dimension, whether or not the task involves prediction for manipulation or ambulatory motion can have a significant impact on the prediction accuracy of a given method. In fact, some methods were specifically designed in the context of manipulation motions (e.g., [98], [102], [117]) while others are framed specifically for ambulatory motion (e.g., [34], [97], [163], [67]). As shown by the fact that the technique described in [117] was used for ambulatory motion as well [145], however,

the context in which a technique is developed or framed is not always indicative of its utility in other scenarios.

Based on how many answers to key questions need to be known in order to select an appropriate predictor, it becomes apparent that it is not straightforward for a potential user of these technologies to make this determination. As can be seen, these questions are difficult to answer and often do not have clear, conclusive answers without performing some additional analysis. In fact, many potential users of these technologies might not even know what questions need to be asked in the first place when trying to select one of the many, vastly different human motion prediction techniques. Furthermore, it is possible that no single prediction method is sufficient in a given scenario, and that a combination of predictors would perform better.

These facts provide strong support and motivation for the first drive of this thesis in developing a framework for automatic formulation of a high-performing predictor out of individual component methods based on given task-data, as was introduced in 1.1. By combining complementary predictors based on their relative performances on data from a given task, this approach allows for implicitly answering the aforementioned questions and selecting the most appropriate and high-performing predictors that make use of the most relevant and useful features for a given scenario. Importantly, this can be done without expert knowledge in the field of human motion prediction, opening the door for more widespread use of prediction in real-world HRI scenarios. Since, as discussed in Section 2.6.1, human motion-prediction is an effective tool for providing for safe and efficient human-robot interaction, facilitating its use can have a significant impact on the growth and expansion of human-robot interaction in many domains.

Deficiencies in Partial Trajectory Alignment

As discussed in Section 1.2, partial trajectory alignment is a key enabling technology for human motion prediction. In fact, as long as a representative trajectory for a given action is well defined, any of the aforementioned action prediction techniques can be adapted for human motion prediction with the use of partial trajectory align-

ment. One well-known and widely used full trajectory alignment method, originally developed for speech recognition, is Dynamic Time Warping (DTW) [128]. DTW uses dynamic programming to identify an optimal time alignment (i.e., *warp path*) between two time series such that the distance between the aligned time series is minimized.

An online variant of DTW (O-DTW) that allows for partial trajectory alignment was introduced in [28]. In this method the values of the warp matrix are computed iteratively as new data is received, with bounds on which values should be computed defined by parameters. The direction of the search is adjusted based on the best alignment up to that point. The bounded warp matrix computations are done with the standard DTW formulation, but using only values of the warp matrix computed in previous iterations. This formulation allows for efficient execution capable of supporting online alignment. This specific approach has been used in domains such as guitar tab score following [100] and trajectory alignment in the context of motion prediction [117]. Several other similar online alignment algorithms based on the DTW formulation have been proposed, including in contexts like human nod detection [153], post-stroke rehabilitation [144], detecting specific motions within a long trajectory during physical therapy [159] and prediction of vehicle motion [114].

One of the problems faced by these types of algorithms is the presence of outliers in the time series being aligned. One approach explored in literature designed specifically for dealing with this problem is Optimal Sequence Bijection (OSB) [92]. In this method, the disturbances caused by outliers are ameliorated by allowing the algorithm to skip points entirely during alignment. This is done by introducing a *jump cost* that penalizes such skips proportionally to how many points are skipped in a row. Online performance is ensured by introducing parameters that limit the warp window size and maximum skip length.

While all of these online partial trajectory alignment techniques tend to work in the domains they are originally designed for, they all contain one fundamental problem with respect to alignment of human motion trajectories. Namely, the distance functions used in these methods all rely on the DTW formulation which uses the

Euclidean distance between aligned points to find the optimal alignment. The reason why this is problematic is that human motion trajectories can contain overlapping regions as well as temporary pauses or stops. These qualities result in many points in the trajectories that are close together in Euclidean space, but can actually be at very different times in the trajectory. Figure 2-8 depicts the problem that can arise with such a formulation.

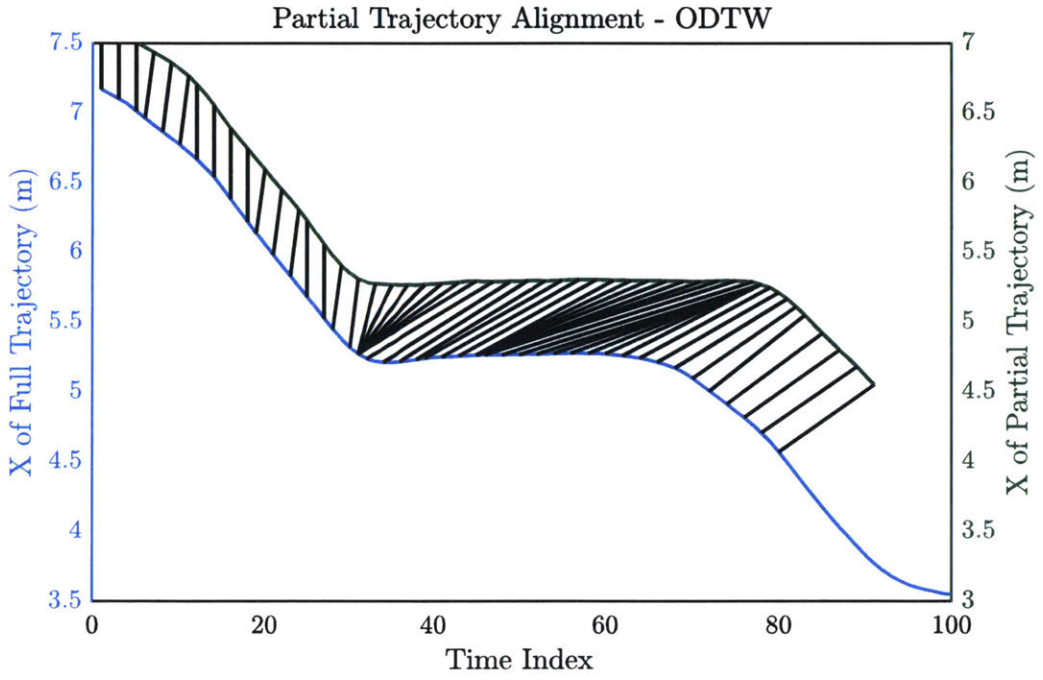


Figure 2-8: An O-DTW partial trajectory alignment between a full trajectory (blue, left y-axis) and a partial trajectory (green, right y-axis). Note the collapse of many points on the representative trajectory onto single points on the full trajectory at time indices t = 31 and t = 46 and the inverse at time index t = 78 of the partial trajectory.

In this figure, we see a representative trajectory (blue) and a partial trajectory of that same motion (green) taken from a human walking motion dataset. The black lines shows sets of aligned points found with O-DTW. Note that the two trajectories are actually very close to each other in Euclidean space, but they are visually separated to show the alignment by shifting the y-axis of the partial trajectory by 0.5m. Due to a pause in the motion, and a slight offset of the partial trajectory, a single point on the full trajectory (t = 31) is the closest point to 15 points in the partial

trajectory (t=32 to t=46), causing O-DTW to map these points together. A similar effect is seen at t = 46. The opposite happens at time t = 78, where many points from the full trajectory get mapped onto a single point on the partial trajectory.

This type of collapse of many points onto one caused by optimizing strictly for Euclidean distance is undesirable, as it does not properly map progress through the trajectory. As discussed in Section 1.2, when partial trajectory alignment is used in the context of human motion prediction, this type of alignment can lead to large prediction errors when attempting to select an appropriate point on the representative trajectory at a queried prediction time horizon. Consequently, this lack of robustness to the common trajectory qualities of incorporating stops and overlaps negatively impacts the robustness of prediction as well. This motivates the development of a partial trajectory alignment method that can accommodate these trajectory types, which is the topic of Chapter 4.

Chapter 3

Developing a General Framework for Robust and Accurate Human Motion Prediction

3.1 Introduction

Accurate and robust prediction of human prediction is a key technology needed for enabling more widespread use of human-robot interaction. As I discussed in Section 2.6.1, predicting where co-located humans will walk or reach to in the future holds a lot of potential in allowing the collaboration or co-navigation to be safe, comfortable, and efficient. These benefits, derived from accurate and robust prediction, have the potential to aid virtually all types of human-robot interaction. As I discussed in Section 2.6.2, however, accurate and robust prediction is difficult to achieve in practice.

The first of the main reasons, discussed in the first subsection of Section 2.6.2, is the difficulty in selecting appropriate predictors for a given task. From the perspective of someone trying to deploy a robot that can collaborate or co-navigate with people in a shared space, it is difficult to decide what type of predictor to use; without expert knowledge in the field of human motion prediction its difficult to even figure

out what the options are. Even with expert knowledge, the answer is not always obvious, as differences in aspects like the types of motions, amount of noise in human tracking, length and duration of trajectories, or discernably of actions can significantly affect the effectiveness of the various predictors. Furthermore, accurate predictions are necessary over both short and long time horizons. Short-term predictions are critical for effective collision avoidance, as the robot must have accurate knowledge of where a person will be in the immediate future in order to stop or execute an evasive motion. While a safety system can serve as a fallback to prevent collisions, it is not a sufficient solution for ensuring that interaction feels safe and comfortable for human workers [88] – an essential requirement for human-robot interaction [14, 9, 135]. As efficiency is also imperative in many domains, there is also a need for accurate long-term predictions. These predictions, when combined with an appropriate planner, allow the robot to make intelligent decisions about how to move toward its own goal in a manner that minimizes interference with humans. As predictors vary in performance as a function of the prediction time horizon, a combination of predictors could be useful.

The above drawbacks limit the practical usefulness of prior approaches of human motion prediction and hinder deployment in real-world applications. In this chapter, I introduce a proposed solution to this problem called the "Multiple-Predictor System" (MPS), a general framework designed for synthesizing a high-performing predictor from a combination of complementary prediction methods. The MPS is a data-driven approach that, based on a variety of data encoding how a person moves within a shared workspace and how he or she performs tasks, will automatically select a favorable combination of prediction methods to accurately predict human occupancy at various future time frames. Given a library of implemented prediction methods, this approach offloads the task of predictor selection and parameter tuning from the user and builds a high-performing predictor that accommodates and exploits the qualities of the motions and task for which predictions are to be made.

Through an evaluation on real human motion data, I show that the MPS automatically learns to exploit the complementary strengths of each component method

to form a combined predictor that outperforms the individual, baseline methods. By performing the evaluation on three different human motion datasets that differ in several key qualities, I provide strong support for the claim that my approach is robust to different types of tasks and motions. Next, I provide further support for the robustness of our method by generating variants of the tasks by forming modified datasets that differ with respect to key task parameters.

3.2 Human Motion Datasets

In order to evaluate the robustness of my approach to different tasks and motions, I collected three distinct human motion datasets that differ in several key qualities. These qualities include those of the motion, such as whether the motion is ambulatory or manipulation and how many degrees of freedom are used, task qualities, such as how many distinct actions there are and how long they last, and sensing qualities, such as whether the sensors are onboard or external. In this section, I describe each of these datasets in detail, discuss the different task and motion qualities, and describe the dataset variants there were generated for the evaluations.

3.2.1 Tabletop Factory Manipulation Motions

The first human motion dataset is a tabletop reaching task in a simulated factory setting (this dataset will be referred to as "TF" in the remainder of this thesis). This dataset was obtained from a previous human factors study presented in [88]. In this prior study, 20 participants collaborated with an ABB IRB-120 industrial robot arm on a tabletop task that required participants to place bolts at designated locations while the robot assisted the person by pretending to apply a sealant over the bolts. This setup is depicted in Figure 3-1. The dimensions of the white foam board in which the bolts were placed (see figure) are 1.5m by 0.5m, and the bolt placement locations are 14cm apart.

The three-dimensional position of the participants' wrist was tracked using a PhaseSpace motion capture system while the person reached between the bolt pickup lo-

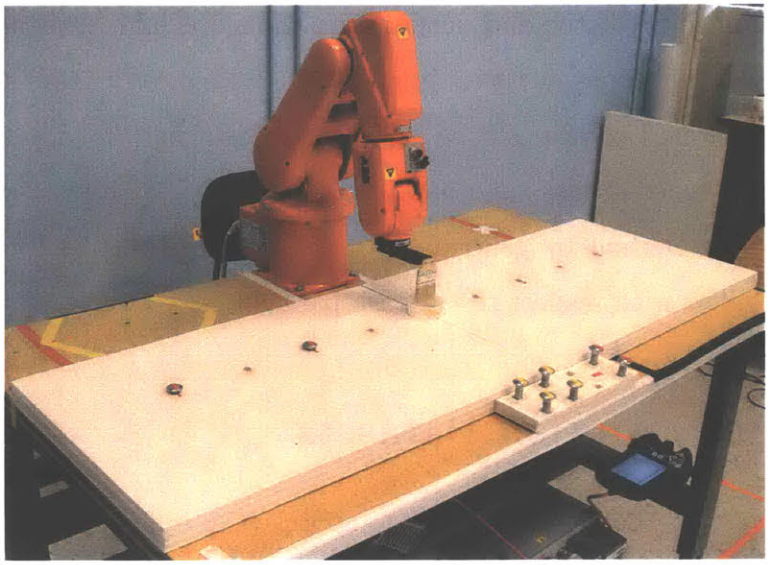


Figure 3-1: The physical setup used to collect the TF dataset. The eight bolts were picked up from the holder and placed in the eight holes in the white foam board (which was 1.5m by 0.5m). Figure originally from [88].

cation and designated placement locations. The data was collected at an average rate of 140Hz and downsampled to 50Hz. In this task each, action was designated as a reach starting at the bolt holder to one of the bolt locations, twisting the bolt to align it with the marker on the table, and then reaching back toward the bolt holder. The average action duration is 4.41 $\pm$ 1.00s, with 1.30 $\pm$ 0.32s spent in motion and 3.10 $\pm$ 0.99s spent twisting the bolt.

The robot operated in one of two motion planning modes: a human-aware mode in which the robot selected paths that avoided portions of the shared workspace that the human was expected to occupy, and a standard mode in which the robot selected the quickest path to its goals without reasoning on where the human would be next. Each participant performed the task with both robot types, and within each task performed eight bolt placements in a preset sequence. A video depicting sample task executions of the human-aware and standard modes is available at the following link: http://youtu.be/Dk5XVQBDJpU.

3.2.2 Automotive Assembly Ambulatory Motions[1].

The second dataset is of walking motions in an analogue automotive assembly task (this dataset will be referred to as "AA" in the remainder of this thesis). In this task, a robot with single axis mobility (moving on an industrial rail) moves between various workstations and parts depots to deliver tools and components to people at the workstations. While the robot is performing its tasks, a person enters the workspace and takes one of the four trajectories depicted in Figure 3-2. The "pause" symbols in the figure represent places where the human would pause for $\approx$ 3 seconds before continuing along their trajectory, which is intended to simulate a worker stopping in order to perform a task (e.g., picking up a tool or reading a value from a monitor). The dimensions of the workspace are approximately 10 m by 3.5 m. Two participants performed each of the four motions 10 times each (for a total of 20 demonstrations per action), while the 2D position and orientation of their head was recorded at a rate of 120 Hz with a Vicon motion capture system. The trajectories were then downsampled to 10 Hz. The average action duration in this dataset is 16.05 $\pm$ 2.41s, with 13.56 $\pm$ 2.10s spent in motion and 2.49 $\pm$ 1.11s waiting.

3.2.3 Analogue Space Motions

The final human motion dataset is of analogue space motions meant to emulate an astronaut's motion in a microgravity environment, such as the International Space Station (this dataset will be referred to as "SSM" in the remainder of this thesis). These motions were collected using a prototype of Astrobee, a free-flying robot designed for operation within the International Space Station [15]. The dataset was recorded with Astrobee's camera and depth sensors, which were used to compute the relative position of a human with respect to the robot at a rate of 5Hz. A custom detection and tracking algorithm based on the OpenCV implementation of Haar Feature-based Cascade Classifiers was used for this task.

[1]This dataset was collected as part of a collaborative project published in [147]

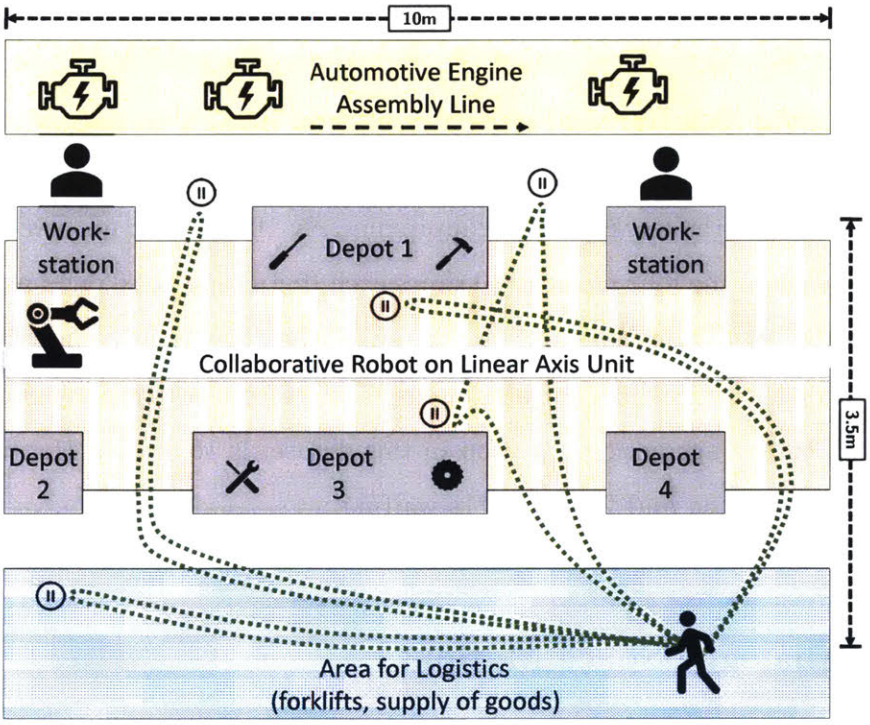


Figure 3-2: Schematic of the analogue factory workspace layout used to collect the AA dataset. The robot moves along a linear axis delivering parts between depots and workstations, while a person moves along one of the paths shown in green. The orange box depicts the shared workspace. Figure originally from [147]

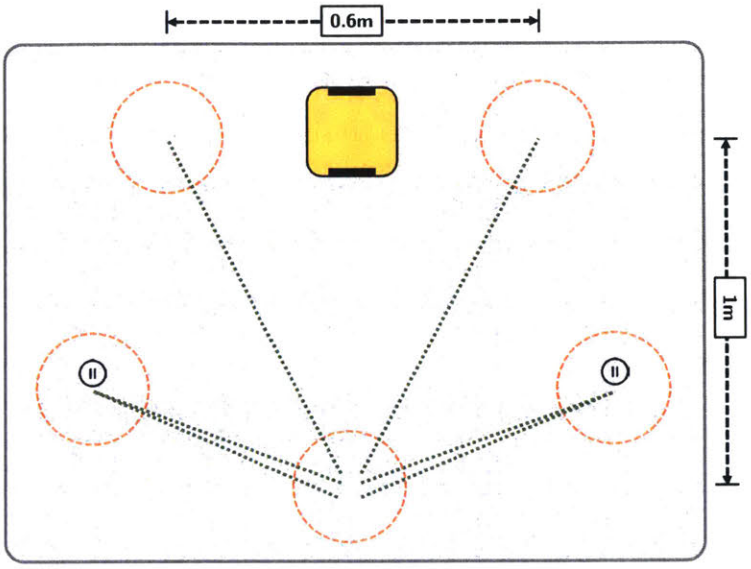


Figure 3-3: Schematic of the possible motions in the analogue space motion dataset. The person moves along the lines shown in green around the Astrobee robot. The orange circles depict the regions containing possible start and end points for each motion.

As the position between the human and robot is relative due to the use of onboard sensors, we simulated an astronaut "flying" near the robot by commanding the robot to glide along different paths on a specialized testbed in the Granite Lab at NASA Ames Research Center. The physical setup used to collect the SSM dataset is depicted in Figure 3-3. The left side of the figure depicts Astrobee and the testbed at the Granite Lab, while the right side of the figure shows a top-down schematic of the motions. Due to the constrained size of the testbed, the area in which Astrobee moved was approximately 1m by 1m.

The dataset consists of 19 examples of four distinct motions collected with the help of three participants. Two of these motions simulated the human moving past the robot on either side, while the other two simulated lateral motions (one to either side). The lateral motions contained a pause of 1-2s followed by a return to the starting location. The average action duration is 22.23 $\pm$ 5.23s, with 18.37 $\pm$ 2.25s spent in motion and 7.73 $\pm$ 1.81s waiting. (Note that the average time spent stopped is longer than the commanded 1-2s due to delays in execution of successive motion commands

issued to Astrobee's flight software. This was caused by making the position tolerance of the controller too precise for the test setup.)

One of the goals of this dataset was to simulate more general passing and lateral movements instead of defining strict paths. In order to support this goal, the start and end points of the trajectories were randomized, adding a random displacement of up to 10cm from the nominal positions (dashed circles in Figure 3-3).

3.2.4 Key Dataset Qualities for Evaluation of Robustness

In order to effectively assess robustness of the MPS to various types of tasks and motions, the datasets used for evaluation were selected with the goal of differentiation in as many qualities as possible. The datasets differ across three main categories of qualities: motion, task, and sensing. In this section I discuss the key differences in these categories with respect to assessing robustness of our methods, which are summarized in Table 3.1.

From this table, one can see that the three datasets differ significantly in terms of motion qualities. As mentioned in the preceding sections, one of the most significant distinguishing factors between the datasets is that the TF dataset deals with manipulation motion in three predicted of freedom, while the AA and SSM datasets deal with ambulatory motions in two degrees of freedom (note: while in the AA dataset head orientation was also tracked, prediction was made only in terms of the two position degrees of freedom). The type of ambulatory motion in the AA and SSM datasets is different, however, as one involves walking, while the other involves an emulated glide in a microgravity environment. As a result, some features that are useful for predicting walking motions (especially biomechanical indicators) are not applicable for the SSM dataset.

Another key difference in terms of the motion qualities are the speeds and accelerations. Note that while the AA dataset has the highest average speed when in motion, the TF dataset contains the highest accelerations. The SSM dataset, on the other hand, contains motions of significantly lower speed and acceleration than the other two. These qualities can affect the relative performances of methods that, for

		Dataset		
		TF	**AA**	**SSM**
Human Motion Qualities	Ambulatory vs Manipulation	Manipulation	Ambulatory (Walking)	Ambulatory (Gliding)
	Degrees of Freedom	3	2	2
	Acceleration	High	Low	Very Low
	Speed (Motion)	$0.58 \pm 0.40\frac{m}{s}$	$0.90 \pm 0.30\frac{m}{s}$	$0.05 \pm 0.02\frac{m}{s}$
	Speed (Stopped)	$0.06 \pm 0.05\frac{m}{s}$	$0.06 \pm 0.05\frac{m}{s}$	$0.01 \pm 0.01\frac{m}{s}$
Task Qualities	Type of Interaction	Collaboration	Co-Navigation	Co-Navigation
	Number of Actions	8	4	4
	Goal Location Spacing	0.14m	1.5-5.5m	0.6m
	Action Duration	4.41 ± 1.00s	16.05 ± 2.41s	22.23 ± 5.23s
	Motion Duration	1.30 ± 0.32s	13.60 ± 2.10s	18.37 ± 2.25s
	Stopped Duration	3.10 ± 0.99s	2.50 ± 1.11s	7.73 ± 1.81s
Sensing	Onboard vs External	External	External	Onboard
	Sensor Type	Motion Capture	Motion Capture	RGB Camera and TOF Depth Sensor
	Tracking Noise	Low	Low	High
	Data Rate	50Hz	10Hz	5Hz

Table 3.1: Comparison of the various motion, task, and sensing qualities for the three human motion datasets.

example, use estimates of velocity as features. The low acceleration and speed combined with the smooth, gliding motion in the SSM dataset, for example, might make it easier to estimate velocities accurately. This effect could be negated, however, due to differences in tracking noise (more on this below).

When a person pauses their motion during a trajectory, the speed computed from position data will generally not be exactly zero due factors like small movements of the human and sensor noise. Consequently, it is also useful to characterize the average speed while in the "stopped" regions of the trajectories. Interestingly, the TF and AA datasets have nearly identical speed in stop regions despite one dataset being ambulatory and the other being manipulation. As can be seen in Table 3.1, the SSM dataset has a much lower mean speed in stop regions.

The datasets also have substantial differences in terms of task qualities. First, the task in the TF dataset is a collaborative task, while in the other two datasets the human and robot are simply sharing the same space while performing independent tasks. The number of possible actions is also quite different, with the TF dataset having twice as many possible human tasks. The larger number of actions in the TF dataset, combined with the smallest spacing between the goal locations of these tasks, has the potential to make goal-based action prediction more challenging. Conversely, the large distances between goal locations and smaller number of actions should make this task easier in the AA and SSM datasets. The accuracy of goal-based motion prediction, however, also depends on the variance in the observed motions for each action, so this trend is not guaranteed. As mentioned in Section 3.2.3, the SSM dataset was collected with randomization in the start and end positions, which can reduce the performance of certain goal-based predictors.

There are also substantial differences in the durations of actions in the three datasets. The actions in the TF dataset are much shorter than those of the other two datasets. Furthermore, another quality unique to the TF dataset is that the time spent stopped accounts for a majority of the action duration. It is also notable that the variance in action duration as a percentage of the mean duration is far lower in the AA dataset than in the others. These differences in the means and

variances of action durations have substantial implications on the suitability of predictors at various time horizons. The significant amount of time spent stopped in the actions of the TF dataset can lead to large errors when using goal-based prediction if the method of partial trajectory alignment is unable to properly accommodate such stops, as discussed in Section 2.6.2. Furthermore, the long durations of the AA and SSM datasets motivate the use of long-term predictors, but these predictors are not necessarily the best choice across all possible time horizons of interest, potentially making a combination of predictors especially useful for these datasets.

Finally, there are also several key differences in the sensing qualities of the datasets. For example, in both the TF and AA datasets a low-noise, external motion capture system was utilized for human position tracking. In the SSM dataset, on the other hand, in order to improve the fidelity of the emulated data, only onboard sensors of the Astrobee robot were used. The custom human detection and tracking framework using these sensors produced far more noisy human tracking data at a much lower rate. These differences in the sensing can have a very significant impact on prediction quality, affecting both goal-based prediction classification accuracy as well as any method that computes velocity estimates.

Overall, there are substantial differences along many key dimensions of motion, task, and sensing qualities of the three human motion datasets. These differences allow for a more reliable assessment of the robustness of the developed methods, as the differences in these qualities can, to varying degrees, impact the accuracy of the various types of human motion prediction. The combinations of these effects can make it difficult to ascertain what prediction methods are most suitable for each dataset, providing further support for the developed method of automatic synthesis of a high-performing predictor.

3.2.5 Dataset Variants

One useful quality of our three datasets is that the trajectory of each action within each dataset originates at the same start location. As a result, it is possible to create new variations of the task in which the order of the actions is altered or only a subset of

Data Set	Actions Used	Action Order	Robot Mode
1	All	1,2,3,4,5,6,7,8	Human-Aware
2	1,4,5,8	1,4,5,8	Human-Aware
3	All	Random	Human-Aware
4	1,4,5,8	Random	Human-Aware
5	All	25% Random	Human-Aware
6	1,4,5,8	25% Random	Human-Aware
7	All	1,2,3,4,5,6,7,8	Standard
8	1,4,5,8	1,4,5,8	Standard
9	All	Random	Standard
10	1,4,5,8	Random	Standard
11	All	25% Random	Standard
12	1,4,5,8	25% Random	Standard

Table 3.2: Dataset variants generated using the TF dataset.

the actions is performed. These qualities were exploited in order to generate variants of the originally collected datasets to further aid in assessing the robustness of the developed techniques.

Starting with the TF dataset, 12 distinct variants were generated. First, we manipulated the action order by drawing the next action from a uniform distribution over the set of actions not yet performed in a given sequence, for either the data from all 20 subjects, 25% of the subjects, or for none of the subjects (causing the action sequence to be identical for all demonstrations). We also manipulated the number of actions to be either the full set of eight original actions or a subset of four actions with goal locations spaced roughly equally apart from one another. Finally, we also used whether the data came from a "standard" or "human-aware" execution. The goal of generating these variants was to evaluate, while holding all other properties constant, how well our method generalizes across the spectrum of more- and less-structured tasks, as well as tasks with several nearby goal locations versus fewer, more separate goal locations. A summary of these data sets is depicted in Table 3.2.

Similar dataset variants were also created for the AA and SSM motion datasets. In the case of these two datasets, the number of actions was already low, and the goal

Data Set	Actions Used	Action Order
1	All	1,2,3,4
2	All	25% Random
3	All	Random

Table 3.3: Dataset variants generated using the AA and SSM datasets.

locations were well-separated (see Table 3.1 and Figures 3-2 and 3-3), and so reducing the number of actions would not provide any useful differentiation. Furthermore, these datasets did not contain different modes like the "human-aware" and "standard" modes in the TF dataset. As a result, for the AA and SSM datasets, the dataset variants were based only on altering action order. The actions in the AA and SSM datasets, however, were not collected in any particular order. Consequently, the example trajectories were randomly split into sets such that each set contained one example of each task. The order of the actions within each set was then assigned based on the condition for that dataset variant. The resulting dataset variants are shown in Table 3.3.

3.3 The Multiple-Predictor System

Here, I describe the Multiple-Predictor System (MPS), a robust, data-driven, general framework for combining various human motion prediction methods together in order to synthesize a high-performing predictor [2]. After describing the method in detail, I discuss the specific proof-of-concept implementation of this framework, specifically developed to assess its robustness with the human motion datasets described in Section 3.2.

[2]The Multiple-Predictor System was initially published in the proceedings of the 2017 International Conference on Robotics and Automation [91]. The citation is as follows:
Przemyslaw A. Lasota and Julie A. Shah. A Multiple-Predictor Approach to Human Motion Prediction. In Proc. International Conference on Robotics and Automation (ICRA), pages 2300-2307. IEEE, 2017.

3.3.1 Overview

An outline of the system architecture of the Multiple-Predictor System is depicted in Figure 3-4. The synthesis of the MPS from a set of various predictors (each based on different methods and features) is composed of two steps. First, a subset of the data, $\mathcal{D}_{Train}$, is used to learn the parameters of each individual prediction method. Next, using the parameters found in the first step, each prediction method is used to compute predictions on a second subset of the data, $\mathcal{D}_{ModelSelection}$. In this step, for each trajectory in $\mathcal{D}_{ModelSelection}$ and for each prediction time horizon of interest ΔT, each predictor is used to generate estimates of future positions denoted as $\widehat{\boldsymbol{x}}_{t+\Delta T}$. The prediction results from each method are then used to compute weights for each predictor that are in turn used to define the rules used by the multiple-predictor system to select which prediction methods should be used.

As I described in Section 2.5.3, there exists a large variety of human motion prediction methods that work in fundamentally different ways and reason on a wide variety of potential features. A key insight that is considered in my approach is that, as a result of these key differences, the relative accuracies of different predictors can change as a function of prediction time horizon. Goal-based methods that perform an alignment to a representative trajectory, for example, might produce relatively poor short-term predictions if the current position does not match the representative trajectory well. Potential-field or biomechanical indicator methods, on the other hand, might work very well for short time horizons, but their performance might degrade for longer time horizons. Furthermore, for a given prediction method, the same set of parameter values that leads to accurate predictions at shorter time horizons might not be the best set of parameter values for making predictions at longer time horizons. Consequently, both the individual predictor parameters and the predictor weights are learned as a function of time horizon.

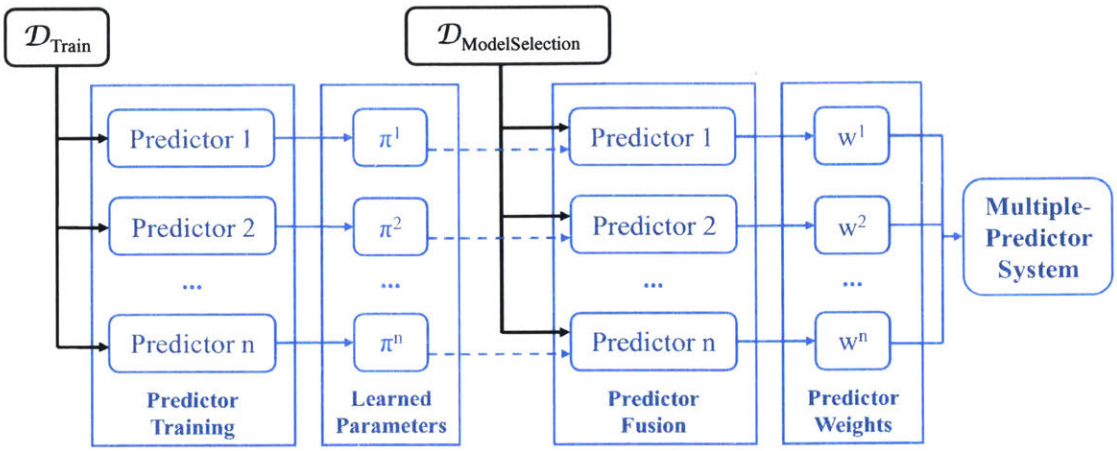


Figure 3-4: An outline of the process of forming the multiple-predictor system.

3.3.2 Training Individual Prediction Components

The first step toward formulating the multiple-predictor system involves learning the parameters of each individual prediction method. Given n available prediction methods, for each method $i = 1 \ldots n$, we define the set of parameters of a method as $\mathcal{Z}^i = \{z_1^i \ldots z_{m^i}^i\}$, where m^i is the number of tunable parameters for method i. Constraining the possible values of individual parameters to given discrete sets, the goal is to identify a set of assignments to each z_j^i that minimizes the training error of each prediction method.

As mentioned in the previous section, the accuracy of a predictor can vary significantly as a function of prediction time horizon. Consider a set of discrete prediction time horizons of interest of length k, defined as $\mathcal{H}$. We denote the individual time horizon values as h_p, where $p = 1 \ldots k$. The objective, therefore, is to find the most suitable parameter value assignment $\mathcal{Z}^i$ for each time horizon in $\mathcal{H}$. If the overall training error of method i given time horizon h_p, parameter assignments $\mathcal{Z}^i$, and input data D is defined to be E^i, the parameter assignment from specified discrete sets that achieves the highest performance for that method and time horizon is defined as follows:

$$\pi_p^i = \underset{\mathcal{Z}^i}{argmin}\, E^i(h_p, D; \mathcal{Z}^i) \tag{3.1}$$

The training process, therefore, results in $k \times n$ set assignments π_p^i. Defining d as the individual trajectories contained in D and $T_{i,d}^*$ as the set of time steps of trajectory d for which a prediction can be produced by predictor i, the training error E^i is defined as the mean value of these prediction errors:

$$E^i(h_p, D; \mathcal{Z}^i) = \frac{1}{|D|} \sum_{d \in D} \frac{1}{|T_{i,d}^*|} \sum_{t \in T_{i,d}^*} \| (\widehat{\boldsymbol{x}}_{t+h_p})^i - \boldsymbol{x}_{t+h_p} \| \tag{3.2}$$

$(\widehat{\boldsymbol{x}}_{t+h_p})^i$ is the position prediction of method i, which is a function of the parameter values $\mathcal{Z}^i$. The need for defining $T_{i,d}^*$ arises from the fact that not all prediction methods can make predictions at all time steps of a trajectory. Some prediction methods require data from some number of time steps to be observed before a prediction can be made or cannot make a prediction beyond the end of the current action based on a partial trajectory alignment, for example.

3.3.3 Predictor Fusion

Once the sets of parameters as a function of prediction time horizon for each individual method are learned during the training phase, the next step in formulating the multiple-predictor system involves determining how to most effectively combine the outputs of these different methods.

When establishing the combination of position predictors, it is necessary to combine the output of several independent learners. Furthermore, the output of the predictors is the future position $\widehat{\boldsymbol{x}}_{t+\Delta T}$, which is a continuous variable, rather than an element of a discrete set of classes. This type of problem lends itself to a class of methods designed for combining the predictions of several experts by evaluating their relative performances and learning which experts tend to produce the best predictions, such as the Hedge Algorithm [45], or the Weighted Average Algorithm [73]

One algorithm within this class — the one selected for the framework — is the Polynomial Weights (PW) algorithm [112]. In this algorithm, each predictor i is treated as an "expert" and is initially assigned a weight $w_0^i = 1$. Then, at each successive time step $t = 1 \ldots T$, an expert i is chosen at random in proportion to its

normalized weight, $w_t^i / \sum_{i=1}^{N} w_t^i$. The weight of each expert is then updated by setting $w_{t+1}^i = w_t^i \cdot (1 - \epsilon L_t^i)$, where $L_t^i \in [0, 1]$ is the prediction loss of expert i at time t and ϵ is the learning rate. As the total number of time steps in all of the trajectories in $\mathcal{D}_{ModelSelection}$ is known, it is possible to set the learning rate ϵ to its optimal value, which is given by $\epsilon = \sqrt{\frac{ln(N)}{T}}$.

Since the output of the multiple-predictor framework is a continuous variable, the prediction error, defined as the distance between the predicted and ground truth positions, is also a continuous variable. Consequently, one of the main advantages of using PW in the framework over other "combination of experts" algorithms is that it includes a continuous loss function. As the PW algorithm expects a loss function limited to the range $[0, 1]$, within the predictor fusion step of the MPS, the prediction errors are normalized with respect to the mean μ and standard deviation σ of the errors encountered in the training phase. Specifically, the loss function is defined as follows:

$$L_t^i = min \left\{ \frac{\| (\widehat{\boldsymbol{x}}_{t+\Delta T})^i - \boldsymbol{x}_{t+\Delta T} \|}{\mu + \sigma}, 1 \right\} \tag{3.3}$$

As mentioned in the previous section, during the training phase, an assignment of individual method parameters for each method and prediction time horizon of interest is computed. Consequently, when forming the combined predictor, the predictor fusion routine runs the PW algorithm for each $h_p \in \mathcal{H}$ independently. Defining the vector of final method weights for each time horizon h_p as $\boldsymbol{W}_p = [w_T^1 \ldots w_T^n]$, the predictor fusion results in k such sets (recall that $|\mathcal{H}| = k$).

Once the final method weight sets $\boldsymbol{W}_p$ are known, the predictor fusion method forms the combined predictor by defining which method should be used at each $h_p \in \mathcal{H}$. In our implementation, the predictor to use at time horizon h_p, defined as i_p^*, is represented as follows:

$$i_p^* = \underset{i}{argmax}(\boldsymbol{W}_p(i)) \tag{3.4}$$

$\boldsymbol{W}_p(i)$ returns the i^{th} element of $\boldsymbol{W}_p$. The reason for using (3.4) as our predictor selection rule, as opposed to selecting a method at random in proportion to the values in $\boldsymbol{W}_p$ as is often done in "combination of experts" methods, is that due to potential safety-critical applications of position prediction (e.g., robot motion planning during human-robot interaction), it is undesirable to query a predictor that returns inaccurate positions, even if that predictor is selected with low probability. The same reasoning is used for not computing a weighted average of the predictions – very poor predictions that are far from the ground truth position, even when given a small weight, can shift the resulting prediction substantially.

3.3.4 Candidate Methods of Human Motion Prediction for Proof-of-Concept Implementation

Multiple classifier systems (MCS) are often designed to incorporate mutually complementary and diverse individual classifiers [156]. Following this guiding principle, a set of three methods of predicting human motion were selected for a proof-of-concept implementation of the multiple-predictor system: velocity-based position projection, time series classification, and action sequence prediction.

The complementary nature of these techniques is derived from their spanning many categories and sub-categories of motion prediction. The first two methods, for example, reason directly on observed human motion, while the third technique utilizes discrete action labels as input. The last two methods focus on predicting action goals as a proxy for predicting position, while the velocity-based position projection technique assesses motion directly without predicting goals. The aim of combining these three techniques is to form a multiple-predictor system capable of producing accurate predictions under a variety of scenarios that no single predictor could adequately address alone.

Velocity-Based Position Projection (VBPP)

The first method of human motion prediction used in the proof-of-concept implementation of the MPS framework is based on projecting the human's current position through an estimate of his or her velocity. Namely, once an estimate of the current velocity, $\widetilde{\boldsymbol{v}}_t$, is obtained, this method computes $\widehat{\boldsymbol{x}}_{t+\Delta T}$, the predicted position in ΔT seconds, by assuming the human will maintain the same velocity for that time period:

$$\widehat{\boldsymbol{x}}_{t+\Delta T} = \boldsymbol{x}_t + \widetilde{\boldsymbol{v}}_t \cdot \Delta T \tag{3.5}$$

To allow for the use of this technique in online motion prediction, $\widetilde{\boldsymbol{v}}_t$ must be computed from observed position data prior to t, $\{\boldsymbol{x}_i\}_{i=1..t}$. This precludes the use of many common filtering techniques that utilize the complete position time series to calculate estimates of velocity at each time t. Furthermore, online prediction necessitates a computationally efficient approach for calculating $\widetilde{\boldsymbol{v}}_t$, so that predictions can be made rapidly as new position data is gathered.

Consequently, the Savitzky-Golay Filter [130] is used to smooth position data and compute velocity estimates. This method works by fitting low-degree polynomials to successive sets of points, and thus does not require the entire trajectory in order to perform smoothing. Furthermore, if the position data is sampled at a uniform rate, there exists an analytical solution to the least-squares fit that can be represented by a set of coefficients. A simple convolution of these pre-computed coefficients with the successive position data can be used to compute a smoothed position signal and its derivatives, rendering the process of estimating velocity from observed position data with this method computationally efficient.

The convolution coefficients of the Savitzky-Golay method are a function of two main parameters. The first is the order of the polynomial to be fit to the data. The best polynomial order to select depends upon the attributes of the position signal, such as the amount of noise present and the sampling rate. The second parameter is the frame size, which defines which portion of the observed position signal is to be used for estimating the current velocity. Namely, for a uniformly sampled signal with

time steps $t = 1 \ldots T$, a frame size of f indicates that, at time t, the set of positions $\{\boldsymbol{x}_i\}_{i=t-f \ldots t}$ will be used for velocity estimation. The polynomial order and frame size are the two parameters that are learned in the first phase of MPS training described in Section 3.3.2.

Time-Series Classification (TSC)

The second implemented motion prediction method is an extension of the goal-based time series classification method developed by Pérez-D'Arpino and Shah [117]. As discussed in Section 2.5.2, this approach uses human demonstrations of motion toward several goal locations to build a library of representative motions based on statistical analysis of tracked degrees of freedom (e.g., positions of various points on the body, head orientation, etc.). Each time step of the motion is encoded as a multivariate Gaussian distribution over these degrees of freedom. The intended goal location is predicted based on early stages of motion by calculating the maximum likelihood that a given partial trajectory belongs to one of the motion classes in the training set.

Depending upon what tracked degrees of freedom are available and the given task, certain degrees of freedom can prove to be more informative than others when it comes to goal prediction with this time series classification method. Some of the tracked degrees of freedom can be completely uninformative and serve to only add noise to the classification process. For example, while the coordinates of a person's hand can certainly provide a useful signal for predicting the goal that person is reaching toward, this may not be the case with other tracked degrees of freedom, such as head orientation. Therefore, in order to find the most informative degrees of freedom to use for goal prediction, this set is learned as part of $\mathcal{Z}^i$ using equations 3.1 and 3.2. Defining the set of all tracked degrees of freedom as $\mathcal{S}$, the set of position coordinates as $\mathcal{X} \subseteq \mathcal{S}$, and the set of the remaining degrees of freedom as $\mathcal{Q} = \mathcal{S} \setminus \mathcal{X}$, the combinations of degrees of freedom that are considered in the training phase, $\mathcal{S}^*$, consist of unions of set $\mathcal{X}$ with each of the elements of the powerset of $\mathcal{Q}$: $\mathcal{S}^* = \{\mathcal{X} \cup q^* \mid q^* \in \mathcal{P}(\mathcal{Q})\}$.

The next step in our extension of the technique by Pérez-D'Arpino and Shah deals with incorporating partial trajectory alignment in order to produce position predictions given the predicted action. Namely, while the original method can predict the goal location a person is walking or reaching toward, it does not directly allow for prediction of that person's position in a given amount of time, $\widehat{\boldsymbol{x}}_{t+\Delta T}$.

The first step of this task is to search the mean trajectory of the predicted goal to identify a suitable representative point that corresponds to the current observed position, $\boldsymbol{x}_t$ (i.e., perform partial trajectory alignment). An implementation of online Dynamic Time Warping was tested for this task, but it led to poorly selected representative points in practice due to the deficiencies of using DTW-based approaches described in Section 2.6.2. Consequently, we implemented an approach based on searching for a nearest-neighbor in a temporal moving window. Let us denote the mean trajectory of goal g as the set of positions $\boldsymbol{x}_t^g$ sampled at uniform time steps $t = 1 \dots T_g$, or $\{\boldsymbol{x}_i^g\}_{i=1\dots T_g}$. Based on the predicted goal g and current time step t, this method selects a representative point $\boldsymbol{x}_\lambda^g$, where the index λ is selected by identifying the point in the mean trajectory that is closest to the current position within a temporal moving window with a size defined by the parameter α:

$$\lambda = \underset{i \in [t-\alpha \dots t+\alpha]}{argmin} \|\boldsymbol{x}_i^g - \boldsymbol{x}_t\| \tag{3.6}$$

The motivation for searching within this moving window is to allow for temporal misalignment between the mean and observed trajectories, but constrained such that erroneous alignments with overlapping trajectory regions are reduced. Higher values of α allow for greater tolerance to temporal misalignment between the mean and currently observed trajectories, but can result in a poor choice of $\boldsymbol{x}_\lambda^g$ and larger alignment errors if the trajectory of the given task contains features like trajectory overlaps or temporary stops. Lower values of α, on the other hand, reduce the likelihood of these types of errors, but cannot accommodate significant temporal misalignments. To find a value that balances these competing goals, we search over a discrete set of candidate

values and select a value of α that minimizes alignment error compared to ground truth alignments found with full DTW.

Once a representative point $\boldsymbol{x}_{\lambda}^{g}$ is selected, the next step is to step forward in the mean trajectory from time step λ until a point that is the desired ΔT ahead is reached. Assuming a sampling rate of f Hz, the predicted position is represented as follows:

$$\widehat{\boldsymbol{x}}_{t+\Delta T} = \boldsymbol{x}_{\lambda+f\cdot\Delta T}^{g} \tag{3.7}$$

Sequence Prediction (SP)

The final motion prediction method implemented in our framework is based on the sequence prediction algorithm developed by Letham et al [93]. One key difference between this method and other sequence prediction approaches is that it reasons on what sets of actions occur before others, and not on the specific order in which these actions occurred. This dramatically reduces the dimensionality of the problem, as it is not necessary to consider all specific orderings of actions, allowing for prediction of sequences with large numbers of possible actions — a desirable ability for a generalizable prediction framework.

Similarly to the time series classification method described in the previous section, the goal of the sequence prediction method is to predict which action a person will take — and, therefore, which goal region he or she will move toward. Unlike the time series classification and velocity-based position projection methods, however, the sequence prediction method reasons on discrete action labels as input instead of working with raw position data.

The implemented sequence prediction approach by Letham et al. incorporates a set of previously observed sequences to learn a set of values, $\lambda_{a,b}$, that describe the relationship between combinations of actions a and b. Large values of $\lambda_{a,b}$ indicate that actions a and b appear together often. The vector of these values, $\boldsymbol{\lambda}$, is then used in a scoring function, which, given a partial sequence of actions and a candidate next action, assigns a score to this action by taking the sum of the λ values that relate the

actions in the partial sequence and the candidate action. Defining the set of actions taken so far up until time t of training sequence i as $x_{i,t}$, the score f of action b is then:

$$f(x_{i,t}, b; \boldsymbol{\lambda}) = \lambda_{\varnothing,b} + \sum_{j=1}^{t} \lambda_{x_{i,j},b} \tag{3.8}$$

The set of parameters $\lambda_{\varnothing,b}$ is needed in order to predict the first action in the sequence. Equation 3.8 is a special case of the "one stage" scoring model described in the publication by Letham et. al. with the constraint that only one action is taken at each time t. Using this scoring function, the candidate action with the highest score is then selected as the action most likely to occur next in the sequence.

The vector $\boldsymbol{\lambda}$ is fit through an optimization of a loss function based on the scoring function and a rule defining which actions should strictly be ranked higher than others based on a given partial training sequence and the known remaining actions. In our implementation, we simply designated that for any given partial training sequence, the next action within that sequence should be ranked higher than all other possible actions.

Using notation from the paper by Letham et al., given a set of training sequences (denoted X_1^m) indexed $i = 1 \ldots m$ of length T_i, with the next action in the training sequence i at time t being $k_{i,t}$, the set of all other actions being $L_{i,t}$, the partial observed sequence i at time t being $x_{i,t}$, and N being the number of possible actions, the loss function we are trying to minimize is represented as follows:

$$R(f, X_1^m; \boldsymbol{\lambda}) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T_i - 1} \frac{1}{T_i} \frac{1}{N-1} \sum_{l \in L_{i,t}} e^{f(x_{i,t}, k_{i,t}; \boldsymbol{\lambda}) - f(x_{i,t}, l; \boldsymbol{\lambda})} + \beta \|\boldsymbol{\lambda}\|_2^2 \tag{3.9}$$

The final term is an l_2-norm regularization scaled by β. The vector $\boldsymbol{\lambda}$ is derived via an unconstrained, convex optimization that minimizes the value of R given the training set of sequences and the scoring function f.

	VBPP	TSC	SP
Goal-Based	No	Yes	Yes
Motion Characteristics	Yes	No	No
Reasons on Position	Yes	Yes	No*
Reasons on Action Sequences	No	No	Yes
Effective Time Horizons	Short	Long	Long

Table 3.4: Comparison of the individual prediction methods used in the proof-of-concept implementation of the MPS.
*While the SP method does not reason on position during goal prediction, it does during the motion prediction stage of the implemented extension described in the previous section.

Once $\boldsymbol{\lambda}$ is computed from the training data, prediction of the next action in a new sequence is performed by identifying the candidate action a with the highest value of $f(x_{i,t}, a; \boldsymbol{\lambda})$. Similarly to the time series classification method, however, the sequence prediction method needed to be extended to predict not only the next goal, but also estimated future positions, $\widehat{\boldsymbol{x}}_{t+\Delta T}$. The same approach as that used for the time series classification method was utilized, incorporating the computed mean trajectories for each action and equations (3.6) and (3.7).

Complementary Features of Candidate Predictors

As mentioned above, when formulating a multiple classifier system, it is desirable to select components that are complementary to one another. The velocity-based position projection, time series classification, and sequence prediction methods were chosen for the proof-of-concept implementation of the MPS precisely due to their complementary nature, as they differ in several key qualities. These differences are summarized in Table 3.4.

The first main difference between the methods lies in that time series classification and series prediction methods are goal-based while the velocity-based position projec-

tion method is based on motion characteristics. While they're both goal-based, the TSC and SP methods reason on drastically different features to make their goal predictions. Namely, the TSC method looks at the actual motion trajectories to build the Bayesian classification model, while the SP method reasons only on observed sequences of actions when making predictions about the next goal.

This quality makes the SP method unique, as it is the only method of the three that does not primarily reason on the observed motion trajectories. Although it does not use the observed trajectories for the main goal-prediction component, the implemented extension that then predicts future positions does, as it performs the partial trajectory alignment based on the current position.

Due to the nature of the three predictors, they also differ substantially in terms of prediction accuracy as a function of tie horizon. Namely, the VBPP method exploits the fact that human motions tend to be smooth without drastic changes in direction, at least at short time scales. This makes it suitable for short term predictions, but, in general, it does not perform well for longer time horizons. The two goal-based methods, on the other hand, use the representative trajectories of the predicted actions, allowing for a more accurate prediction of where the person will be moving to at longer time horizons. If the variance of the trajectories of each action is large, however, matching to a point on the representative trajectory for short time horizons can lead to poor predictions. Whether this is the case or not, and at precisely what time horizon the TSC and SP methods might outperform the VBPP method is highly dependent on the task at hand, further motivating the use of an automatically generated, combined predictor.

3.4 Evaluation

In order to evaluate the relative performances of the individual prediction methods and our combined predictor, I performed a set of leave-one-out cross validations (LOOCV) for each variant of each human motion dataset shown in Tables 3.2 and 3.3.

	TF	AA	SSM
Time Horizon	0.05:0.05:1s	0.1:0.2:6s	0.1:0.2:6s
Window Size α (TSC & SP)	2:2:30	2:2:24	1:1:16
Polynomial Order (VBPP)	1:1:5	1:1:5	1:1:5
Frame size (VBPP)	2:1:10	2:1:10	2:1:10

Table 3.5: Prediction time horizons and parameter ranges for parameter learning used in training of the individual predictors and the MPS.

For each iteration of the cross-validation, one sequence of actions was retained for testing and the remaining data was randomly assigned to $\mathcal{D}_{Train}$ and $\mathcal{D}_{ModelSelection}$, with 70% of the data assigned to the former and 30% to the latter. For the TF dataset, due to 20 subjects performing each of the 20 sets of actions, this split was done by subject ID. In the AA and SSM datasets, since the trajectories were not collected in specific sequences, the cross-validation split was done according to the random assignment of actions into sets, as described in Section 3.2.5.

For each of the iterations of the LOOCV, the mean prediction errors on the trajectories of the test set (in meters) were computed. This was done for the MPS as well as for each of the individual prediction methods, which served as baselines for comparison. The ranges of time horizons considered were based on the typical action durations for each dataset such that there would be a sufficient amount of ground truth data for meaningful evaluation. The time horizon ranges were 0.05s to 1.0s in increments of 0.05s for the TF dataset and 0.1 to 6.0s in increments of 0.2s for the AA and SSM datasets. The discrete sets of parameter values used in the predictor training step of the MPS were selected based on results obtained from tests of the training process such that the best value lies within the given ranges. These ranges are shown in Table 3.5 in the format minimum : increment : maximum.

3.5 Results and Discussion

In this section, I discuss the results of the evaluation of the MPS using the three human motion datasets. First, I discuss the quantitative analysis of prediction accuracy of the

MPS and the individual, baseline predictors for all datasets and their variants. Next, in light of these results, I discuss the qualities of the synthesized MPS predictors, such as the down-selection of the individual predictors, the parameter values learned, and predictor selection as a function of time horizon.

3.5.1 Prediction Accuracy

The prediction accuracy of the individual predictors and the MPS were computed by taking the mean prediction error (in meters) for each trajectory of each dataset. Namely, for each dataset and variant (Tables 3.2 and 3.3), during each iteration of the cross validation, the four predictors were queried for predictions at the sets of time horizons described in Table 3.5. The prediction errors at each of these time horizons were collected for each time step of every trajectory. The mean error for a given predictor and trajectory pair was then computed as the mean of all prediction errors at each time step and time horizon for that trajectory. The overall means for a given predictor and dataset were then computed by averaging the errors across all of the cross-validation iterations. These overall mean errors are summarized in Tables 3.6, 3.7, and 3.8.

As the mean errors of each method evaluated on a specific trajectory are dependent values, I utilized the Friedmann test for statistical analysis. The treatments of the Friedman test are the four predictor types (the three individual methods and the multiple-predictor system). Once the main effect of the prediction method was confirmed, I performed a post-hoc, pairwise comparison with the Wilcoxon signed-rank test. This was done in order to assess whether the reduction in mean prediction error when using the MPS with respect to each of the individual methods was statistically significant. The results of this analysis for each dataset are summarized in the following sections. Note that statistical tests were only performed within each dataset variant, and so differences between the datasets are only reported as trends.

Tabletop Factory Reaching Motions

Overall, for the TF dataset, the MPS outperformed the individual predictors for all 12 dataset variants shown in Table 3.2. The results of the statistical analysis outlined above show that the main effect of predictor type was significant at a confidence level of $p < 0.05$ for all dataset variants. Furthermore, the results show that the mean prediction error of the multiple-predictor system was statistically significantly lower than all three of the individual methods, also at a confidence level of $p < 0.05$, with the exception of dataset 12 (where there is no statistically sigificant difference between the MPS and the SP method).

Data Set	VBPP	TSC	SP	MPS
1	0.076±0.022	0.059±0.033	0.054±0.031	**0.047±0.022***
2	0.076±0.023	0.057±0.034	0.054±0.033	**0.048±0.023***
3	0.076±0.022	0.059±0.033	0.338±0.297	**0.051±0.021***
4	0.076±0.023	0.057±0.035	0.290±0.334	**0.049±0.023***
5	0.076±0.022	0.059±0.033	0.092±0.144	**0.051±0.022***
6	0.076±0.023	0.057±0.035	0.145±0.243	**0.049±0.024***
7	0.105±0.034	0.095±0.054	0.079±0.051	**0.067±0.034***
8	0.105±0.036	0.091±0.054	0.077±0.048	**0.066±0.033***
9	0.105±0.034	0.095±0.053	0.331±0.271	**0.080±0.030***
10	0.105±0.036	0.092±0.055	0.310±0.316	**0.077±0.032***
11	0.105±0.034	0.095±0.053	0.091±0.085	**0.079±0.031***
12	0.105±0.036	0.091±0.054	0.105±0.135	**0.075±0.032**†**

Table 3.6: Mean prediction errors ± standard deviation (meters)
for the variants of the TF dataset
* MPS error lower than all individual methods ($p<0.05$)
† MPS error lower than TSC and VBPP only ($p=0.83$ for SP)

Manipulating the number and order of actions in our dataset variants generated a variety of representative scenarios in which the individual prediction methods achieved differing levels of performance. For example, in scenarios in which only four of the eight actions were taken, the time series classification method generally performed better. This trend can be observed from lower mean errors for dataset variants 2, 4, 6, 8, 10, and 12 compared with dataset variants 1, 3, 5, 7, 9, 11, respectively (see Table

3.6). This is likely due to the fact that there are fewer possible goal regions that are more physically separated from each other, making the learned motion classes more distinct, leading to better prediction performance. While this trend persists across all comparisons of dataset variants with eight and four actions, the magnitude of the difference in error is small. One possible explanation for this is that in the TF dataset, the majority of each trajectory is spent stopped (see Table 3.1), and this stop occurs at the goal location. As a result, the TSC predictor can easily distinguish between the different goals at the point the stop begins, regardless of the number of actions in the dataset. This likely dilutes the effect of the reduced number of actions, which has the strongest impact in the very beginning of each trajectory.

Changing the sequence in which the actions were performed represented another manipulation of the original dataset. In scenarios in which the next action was drawn from a uniform distribution over the set of remaining actions, the sequence prediction method performed poorly compared with scenarios in which the tasks were always completed in the same sequence. This trend can be seen by the much higher mean prediction error (and standard deviation) for the SP method for dataset variants 1, 2, 7, and 8 when compared with 3, 4, 9, and 10, respectively. For example, the mean prediction error from dataset variant 1 increased by a factor of six when the actions were randomized (dataset variant 3) and the standard deviation of the error grew tenfold. These results can also be compared with the performance on dataset variant 5, in which 25% of the sequences were randomized. In this case, the error fell between those of variants 1 and 3, showing that although the prediction performance suffered due to the stochasticity, the technique was still able to find the dominant ordering and make useful predictions when it was encountered.

A related trend that can be observed from the results shown in Table 3.6 is that while the performance of the sequence prediction method is significantly affected by these changes, the performance of the MPS is not as drastically affected. For example, one can see that the mean and standard deviation of the prediction error of the MPS increased by much smaller magnitudes when the action sequences were randomized.

Furthermore, one can see that when the number of randomized sequences increased from 25% to 100%, there is nearly no difference in performance for the MPS.

One can also see an interesting trend in the relative performances of the sequence prediction and time series classification methods. When the sequence order is consistent (dataset variants 1, 2, 7, and 8), the SP method outperforms the TSC method consistently (and by a larger margin for the dataset variants in the standard robot mode). When the sequences are 25% randomized (dataset variants 5, 6, 11, and 12), however, the TSC method begins to outperform the SP method. This effect is even more significant when the sequences are fully randomized (dataset variants 3, 4, 9, and 10). The reason for this difference in performance can likely be attributed to the fact that when the sequence orders are perfectly consistent, the SP method can recognize which action is being performed at a given moment with 100% accuracy immediately after the previous action is completed. The TSC method, on the other hand, reasons on the motion trajectories and therefore needs to observe at least a small portion the motion being executed in order to make an accurate prediction, resulting in a higher prediction error. Once enough stochasticity is introduced into the sequence orderings, however, the performance of the SP method degrades enough that the TSC method beings to outperform it.

Another interesting trend can be seen when comparing the performance of the velocity-based position projection method when using the "human-aware" and "standard" datasets. One can see that in the dataset variants corresponding to the standard mode (variants 7-12), the VBPP method performed worse than for the data collected when the robot was in the human-aware mode (variants 1-6). This suggests that there could be some underlying difference in the participants' motion qualities when interacting with the robot running in these two modes. One plausible explanation is that in the human-aware mode, the robot interfered less with the participants' reaching motions, resulting in more consistent, straight-line motions.

Overall, the key differences in performance of the predictors across the twelve dataset variants discussed above provide strong support for the utility of generating these variants in evaluating the robustness of the MPS. The manipulated quali-

ties have noticeable effects on prediction performance, with several emerging trends. While some of these trends were expected (e.g., the SP method performing poorly when sequence are randomized), the magnitude of the effects and changes in relative performances with other methods could not have been easily predicted without this analysis. Furthermore, some of the emergent trends were not anticipated (e.g., VBPP performance differences for human-aware and standard robot motion).

The appearance of these trends, the difficulty in predicting them and their effects, and the resilience of the MPS to the various dataset manipulations all provide strong support for the robustness of the developed approach. Not only did the MPS outperform the individual prediction methods in all of the dataset variants, the predictor was synthesized automatically without the need for performing this analysis in order to select an appropriate predictor for each dataset variant. Furthermore, the results support the hypothesized notion that in some scenarios a single predictor might not be sufficient, and that a combination of predictors can outperform all individual methods.

Automotive Assembly Ambulatory Motions

The results obtained with the use of the AA dataset show similar trends as those of the TF dataset. As before, the MPS outperforms each of the individual methods in terms of mean alignment error, as can be seen in Table 3.7. The results of the statistical analysis show that the main effect of predictor type was significant at a confidence level of $p < 0.05$ for all dataset variants. The mean error of the MPS was statistically significantly lower for all pairwise comparisons with the exception of dataset variant 2 (25% random action sequence), where the difference between the MPS and SP methods were not significant.

While the MPS performed better than the baselines, the margins between the mean error of the MPS and the second-best predictor are very low. Analyzing the synthesized MPS models in dataset variant 1 (consistent action order) reveals that the VBPP method was selected for time horizons up to 0.7s and the SP method was selected for all time horizons from 0.8s to 6s. Due to the MPS relying on the SP

method for over 85% of the discrete time horizons considered, the overall performances of the MPS and SP methods will tend to be similar, as the performance gained in short time horizons by using the VBPP method is diluted. A similar effect occurs in dataset variant 3 (randomized action order), but in this case the MPS uses the TSC method for all time horizons after 0.7s.

While the overall differences in mean errors are small, the improved performance of the MPS at short time horizons can still be beneficial. As discussed in the introduction of this chapter, accurate predictions at short time horizons are important for collision avoidance. The more accurate the predictions at these short time frames, the better the chance of being able to plan a new motion that avoids collision, or, if there is not enough time to replan, a control-based safety can be triggered in order to minimize potential harm.

Another trend that can be seen in Table 3.7, is that the mean errors of the VBPP and TSC methods stay consistent across all three dataset variants, while the SP method's performance degrades by a large margin. Similarly to the results of the TF dataset, the SP method is shown to produce larger errors with increased randomization of the action sequences. This result makes sense, as the SP method relies on finding patterns in action sequences. While the MPS relies on the SP method in dataset variant 1, its performance only degrades by a small margin as the actions are randomized, once again providing evidence for the benefit of the data-driven predictor selection of the MPS.

Data Set	VBPP	TSC	SP	MPS
1	1.572±0.217	0.346±0.171	0.303±0.120	**0.296±0.112***
2	1.572±0.217	0.343±0.166	0.482±0.695	**0.335±0.159†**
3	1.572±0.217	0.344±0.165	1.493±1.332	**0.337±0.158***

Table 3.7: Mean prediction errors ± standard deviation (meters)
for the variants of the AA dataset
* MPS error lower than all individual methods ($p<0.05$)
† MPS error lower than TSC and VBPP only ($p=0.51$ for SP)

Simulated Space Motions

The results from the SSM dataset indicate similar trends to those of the previous two datasets. Namely, the mean error of the MPS is lower than that of each of the baseline methods for all three dataset variants. The results of the statistical analysis show that the main effect of predictor type was significant at a confidence level of $p < 0.05$ for all dataset variants. The mean error of the MPS was statistically significantly lower for all pairwise comparisons with the exception of dataset variant 2 (25% random action sequence), where the difference between the MPS and SP methods were not significant.

Similarly to the AA dataset results, the VBPP and TSC methods have relatively consistent performance, while the mean error of the SP method grows as action order randomization increases. Once again, due to the MPS taking advantage of the complementary qualities of the predictors, its performance degrades at a much slower rate.

One difference in the results of the SSM dataset when compared to the other two is that the relative performances of the TSC and SP methods for dataset variant 1 (no sequence randomization) is more pronounced. Namely, using the SP method over the TSC method leads to a 32% reduction in error, while this difference is only 8% for the TF dataset and 12% for the AA dataset. This difference in performance can be attributed to the fact that, as described in Section 3.2.3, in the SSM dataset the goal was to simulate more general motions, such as "passing on the right," instead of defining strict paths. As the TSC method attempts to build a multivariate Gaussian distribution for each time step of a representative trajectory for an action, this added variation in the trajectories results in flatter distributions and, therefore, makes it difficult to discern between the different actions.

Another difference in the results of the SSM dataset is the improved performance of the VBPP method in comparison to the TSC and SP methods. In the TF and AA datasets, the VBPP method was typically outperformed by a large margin. In the SSM dataset, on the other hand, the VBPP method outperforms the TSC method

across all dataset variants, and the margin between this method and the SP method for variant 1 is small. The improved performance of the VBPP method can be likely be attributed to several factors. First, the human motions in this dataset are slower and have lower accelerations (see Table 3.1). This leads to lower relative error in velocity estimation. Second, due to the data being designed to emulate space-like motions in microgravity, the paths are all straight lines. These types of motions are very well suited for projecting position forward based on the current velocity, even at longer time horizons.

Data Set	VBPP	TSC	SP	MPS
1	0.066±0.010	0.076±0.038	0.052±0.024	**0.042±0.013***
2	0.066±0.010	0.076±0.038	0.096±0.134	**0.058±0.032**†
3	0.066±0.010	0.076±0.038	0.265±0.242	**0.061±0.021***

Table 3.8: Mean prediction errors ± standard deviation (meters)
for the variants of the SSM dataset
* MPS error lower than all individual methods ($p<0.05$)
† MPS error lower than TSC and VBPP only ($p=0.40$ for SP)

3.5.2 Qualities of Synthesized Predictors

In light of the observed improvements in prediction performance when using the MPS, in this section I analyze and describe the qualities of the synthesized predictors that contributed to this improved performance. This includes intelligent down-selection of predictors and principled assignment of these predictors as a function of prediction time horizon.

The first quality of the synthesized predictors that contributed to the observed improvements in performance was intelligent selection of the individual predictors when synthesizing the MPS. As mentioned in Section 3.5.1, for example, the performance of the MPS was not as negatively affected as the SP method was by the randomization of action sequences. Recall that when the action sequences were consistent, the SP method outperformed the TSC method, and that this effect was reversed otherwise. Consequently, when action sequences were consistent, the SP method received

a higher weight during the model selection step of MPS synthesis, resulting in the MPS being composed of the VBPP and SP methods. When action sequences were not consistent, however, the TSC method's weights increased, resulting in it being selected for the MPS instead.

Consider the plots of mean prediction error as a function of time horizon for consistent and randomized action order in the TF dataset depicted in Figures 3-5 and 3-6, respectively. These figures show that the MPS automatically learned the abovementioned tradeoffs between the TSC and SP methods. Namely, when the action order was consistent (Figure 3-5), the MPS is constructed from the VBPP and SP methods, and when it is randomized (Figure 3-6), the error of the SP method goes up, and the MPS is composed of the VBPP and TSC methods instead. This mechanism allowed for the aforementioned resilience of the MPS to changes in the randomness of sequence ordering.

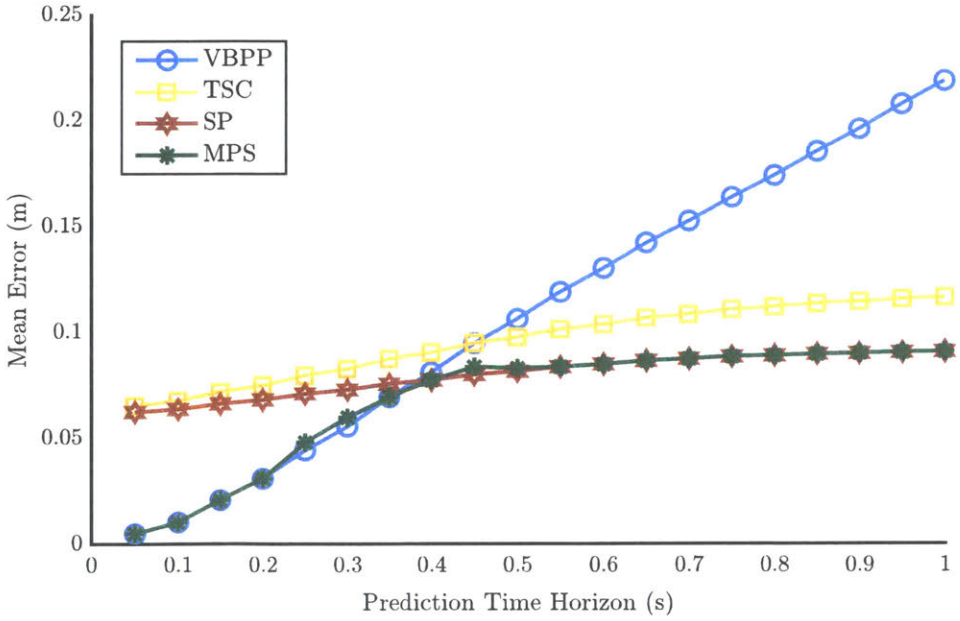


Figure 3-5: Mean prediction errors as a function of time horizon for TF dataset variant 7.

The selection of the velocity-based position projection method as part of the MPS also contributed to the improved prediction performance. As one might anticipate, the VBPP method performs very well at short time horizons, as it does not require

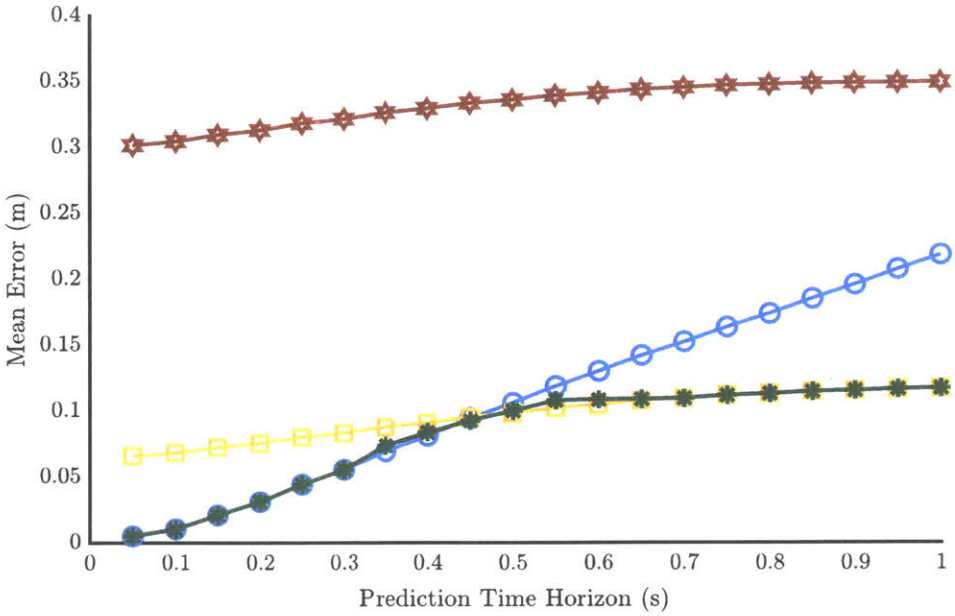


Figure 3-6: Mean prediction errors as a function of time horizon for TF dataset variant 9.

conforming to a representative trajectory like in the TSC and SP methods. This ability of the VBPP to produce low-error predictions at short time horizons led to its automatic selection by the MPS for all datasets and variants.

Besides down-selecting to two of the three methods appropriately, another interesting behavior of the multiple-predictor system is the way in which it combined the selected methods. Consider again the plots of mean prediction error as a function of prediction time horizon for the TF dataset depicted in Figures 3-5 and 3-6. Note that the velocity-based prediction method performed very well at short time horizons (ΔT), but that performance degraded quite rapidly as the time horizon increased. This makes sense, since at larger time horizons, the person might change direction during the time interval in question, resulting in poor prediction and higher mean error.

The time series classification and series prediction methods also exhibited a trend toward decreased performance at higher values of ΔT, but their mean error grew at a slower rate. Furthermore, in cases in which the multiple-predictor system selected these two methods, the mean error at low values of ΔT was higher for these methods

than for the velocity-based method, while this trend was reversed at high values of ΔT. One possible explanation for the relatively poorer performance at low values of ΔT is the variance in the example trajectories: If variance is high enough, the mean trajectory can be quite far from the trajectory being predicted.

While the relatively high error at low values of ΔT is not ideal, the time series and sequence prediction methods' performance degrades slowly with increasing values of ΔT. This is likely due to the fact that when these prediction methods correctly classify the goal the person is reaching toward, the mean trajectory of the motion class yields a much better prediction further into the future than simply assuming the person will continue to move their hand in a straight line.

These attributes create a natural "cross-over point" at which one method becomes superior to the other as a function of prediction time horizon. The plots in Figures 3-5 and 3-6 indicate that the multiple-predictor system was able to learn and exploit this concept automatically.

Comparing this result from the TF dataset to that of the other datasets, however, shows that this "cross-over" can occur at different time horizons depending upon the specific motion being predicted. Consider, for example, dataset variants where the action order was consistent. Figures 3-7 and 3-8 depict the prediction error as a function of time horizon for dataset variant 1 of the Automotive Assembly and Simulated Space Motion datasets, respectively. From these plots and Figure 3-5, one can see that while the cross-over from the VBPP to the SP method occurred at around 0.4s for the TF dataset, it occurred at 0.7s for the AA dataset and 2.5s for the SSM dataset. As one can see, determining exactly where this cross-over point between predictors occurs for different datasets is not straightforward without careful analysis, further supporting our concept of automatic synthesis of the MPS based on given data.

Figure 3-8 also provides further support for the relative changes in performance of the individual methods in the SSM dataset discussed in the previous section and shown in Table 3.8. Namely, the time series classification method performs poorly on this dataset, which is evident from its higher error in comparison to the SP method.

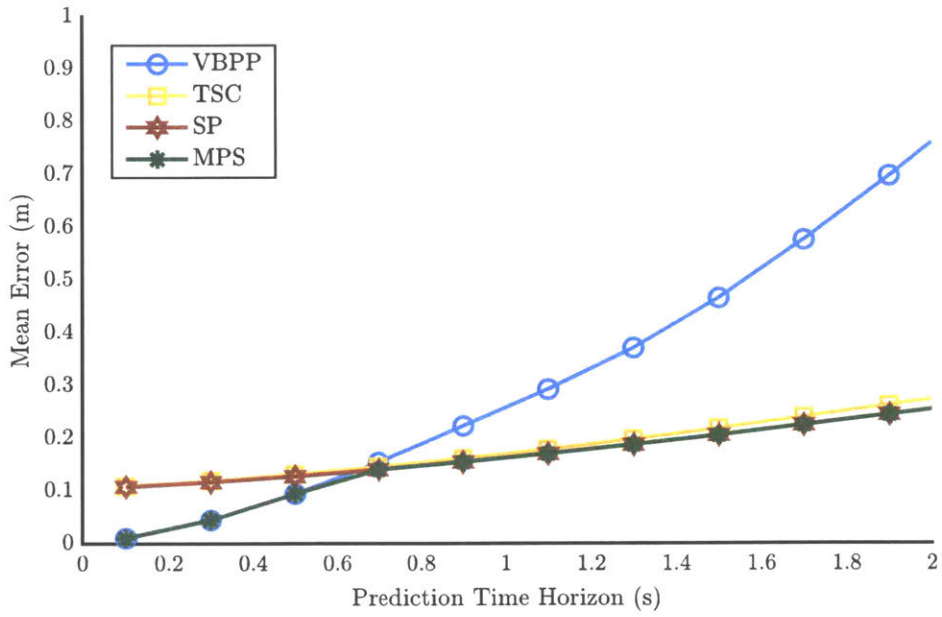


Figure 3-7: Mean prediction errors as a function of time horizon for AA dataset variant 1. Note the plot only depicts time horizons up to 2s in order to better visualize the cross-over from the VBPP method to the SP method.

While the SP method always outperformed the TSC method when the action order is consistent, this difference is very pronounced in this dataset when compared to the TF dataset (Figure 3-5) and AA dataset (Figure 3-7). Second, the plot confirms that the velocity-based position projection method performs better in this dataset, pushing back the cross-over to the SP method to 2.5s.

The decreased performance of the TSC method in the SSM dataset also leads to another unique quality of the synthesized MPS. When the action sequences are randomized, and the MPS switches from using the SP method to the TSC method, the cross-over happens at a much later time horizon. This effect is confirmed in Figure 3-9, which depicts mean prediction error as a function of time horizon for dataset variant 3 of the SSM dataset. The cross-over is pushed forward from 2.5s to nearly 4s. Once again, all of these differences in relative performances of the predictors are automatically captured and exploited by the MPS framework.

Overall, the automatic adjustment in response to the variety of scenarios captured by the three datasets and their variants showcases the ability of the multiple-predictor

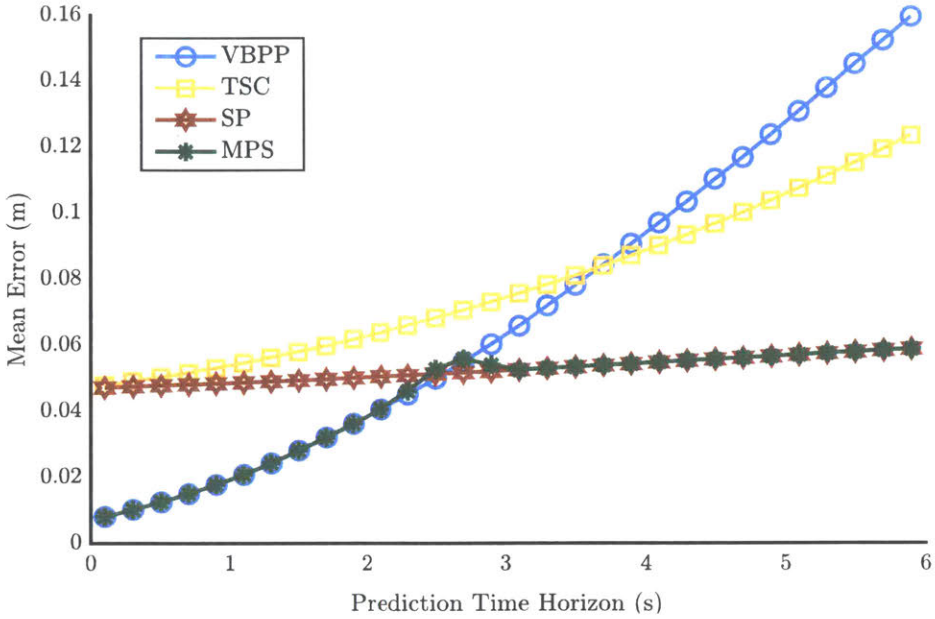


Figure 3-8: Mean prediction errors as a function of time horizon for SSM dataset variant 1.

system to learn from the given training data how to best combine the complementary strengths of several predictors, greatly improving the robustness and generalizability of human motion prediction compared with reliance upon a single prediction method. The MPS achieved this goal quite well, with the mean error of the multiple-predictor system closely tracking the mean errors of the best performing methods at each time horizon. The only points at which the MPS did not select the best performing method was at the transition point between two methods, which is likely due to slight differences between the training and testing data during the leave-one-out cross-validation.

3.6 Conclusions and Future Work

In this chapter, I introduced the Multiple-Predictor System, a data-driven approach to synthesizing a robust, high-performing predictor from a library of individual prediction methods. The developed approach allows for automatic parameter tuning, feature selection, and generation of a combined predictor without requiring expert

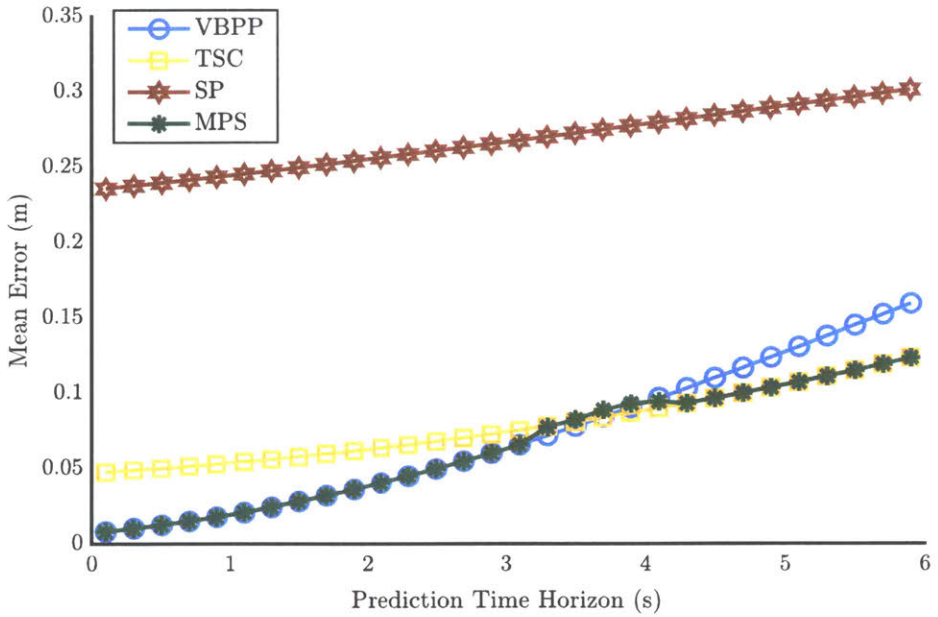


Figure 3-9: Mean prediction errors as a function of time horizon for SSM dataset variant 1.

knowledge, facilitating the use of human motion prediction in human-robot interaction contexts. I evaluated this approach through a proof-of-concept implementation involving three predictors: velocity-based position projection, time series classification, and sequence prediction. With the use of three distinct human motion datasets, and derived dataset variants in which key task qualities were manipulated, I was able to demonstrate the robustness of the developed approach to a variety of key motion, task, and sensing qualities. Overall, the evaluation demonstrated that the MPS outperformed the baseline predictors in a wide variety of scenarios by leveraging the complementary strengths of different prediction methods.

Importantly, the MPS is a general framework, and is not limited to only the three implemented predictors. The benefits of the MPS and its generalizability can be further extended by incorporating new types of predictors, such as potential field approaches for dealing with deviations from a representative trajectory or methods considering social forces if there are multiple humans involved. By incorporating such new predictors as part of the framework, the utility of the MPS can be extended to new domains, with the most appropriate predictors being selected as needed.

One of the key limitations of this approach is that the performance of the synthesized predictor is strongly reliant to the provided training data, and cannot adapt to changes in the task or motions on the fly. If the way in which the task is performed changes over time, the weights learned by the MPS can lead to suboptimal predictor selection. Consequently, one avenue of future work is to make use of the online nature of the Polynomial Weights algorithm in order to continuously update the predictor weights based on relative prediction performances.

Another limitation of the approach is that it currently does not consider how predictor performance changes as a function of what portion of a given action is being executed. Many prediction approaches vary in their performance depending on how much of a motion has been observed. As a result, it would be useful to also incorporate this parameter in the model selection phase, allowing for predictor selection based not only on prediction time horizon, but also estimated activity progress.

Finally, another avenue of future work involves exploring alternative methods of combining the output of the component predictors. In the current framework, the predictor with the highest weight given by the Polynomial Weights algorithm at each time horizon is used for prediction. It might be beneficial, however, to combine the outputs of the predictors based on their relative confidence values. This is especially true near the cross-over points, as the current approach can lead to discontinuities in the predicted path.

Chapter 4

Toward Robust Partial Trajectory Alignment via a Bayesian Estimation Framework

4.1 Introduction

Temporal alignment of trajectories is explored in a wide variety of human-robot interaction domains, including activity detection, gesture recognition, and human intent prediction. Work in the last of these categories includes both action and motion prediction, which, as described in Section 2.5, are key for facilitating safe and efficient HRI. As the goal of prediction is to anticipate future human tasks and motions, temporal alignment in these contexts often deals with reasoning on partially executed trajectories. As a result, partial trajectory alignment is a key enabling technology for accurate and robust human motion prediction.

The problem of partial trajectory alignment, which is distinct from aligning two complete time series, involves identifying a suitable point on the full trajectory that corresponds to the currently observed position. Finding this point allows prediction techniques to estimate the progress through the task, which can then be used, for

example, to query a GMM from a specific time interval [57] or consider a multivariate Gaussian at the best representative time step [117].

To describe the problem of partial trajectory alignment more formally, let us define trajectories as matrices $X \in \mathbb{R}^{N \times T}$, where N is the number of tracked degrees of freedom (e.g., the spatial coordinates of a person's hand) and T is the number of time steps. Each column of X, denoted as $\vec{x}_t$, defines a vector of coordinates at a particular time step $t \in 1 \dots T$, and the trajectory is sampled at some time interval τ. A subset of a trajectory matrix up to some time t is denoted as $X_{[1,t]}$. A partial trajectory for some action, denoted as X^P, is appended with a newly observed set of coordinates, $\vec{x}_t^P$, at each consecutive time step t. Given a complete representative trajectory for the same action, X^R, the objective of online partial trajectory alignment is to identify a *correspondence point* in the representative trajectory, $\vec{x}_{t^*}^R$, that accurately maps to the currently observed position in the partial trajectory, $\vec{x}_t^P$. In other words, the goal is to find a t^* such that X^P maps to the subset $X^R_{[1,t^*]}$. While constructing a mapping between all of the points in $\vec{x}_t^P$ to the points in $X^R_{[1,t^*]}$ can also be considered part of the partial trajectory alignment problem, once a value of $\vec{x}_{t^*}^R$ is estimated, this sub-problem becomes one of a full trajectory alignment between $\vec{x}_t^P$ and $X^R_{[1,t^*]}$. Consequently, they key differentiating problem, which we use in our definition above, is finding the correspondence point.

While several partial time series alignment methods have been introduced, they are generally not suitable for alignment of human motion trajectories. Namely, as described in Section 2.6.2, prior approaches are not resilient to two qualities of human motion trajectories that are often encountered: overlapping regions and temporary stops. Furthermore, prior approaches only provide a single estimate of the correspondence point, but producing a distribution over possible alignments is desirable.

In order to address the drawbacks highlighted above and in support of providing robust human motion prediction, in this chapter, I introduce BEST-PTA (Bayesian ESTimator for Partial Trajectory Alignment), a Bayesian estimation framework composed of a mixture of optimization, supervised learning, and unsupervised learning components that are trained and synthesized based on a given set of example trajecto-

ries. I specifically address the problems of trajectory overlap and temporary stops via segmentation and stop segment detection, and compute and leverage correspondence point priors and a segment classifier to produce the correspondence distribution.

Next, I evaluate the framework against three partial time series alignment baselines, and show that my approach outperforms these baselines across the three human motion datasets introduced in Section 3.2: tabletop factory manipulation motions (TF), automotive assembly ambulatory motions (AA), and simulated space motions (SSM). As was established in Section 3.2.4, these datasets differ across many key motion, task, and sensing qualities, which facilitates the evaluation of the robustness of BEST-PTA as well. As the objective of this thesis is to allow for robust and accurate prediction of human motion, I also evaluate the impact of using the developed alignment technique on prediction with the same three datasets and the Multiple-Predictor System. Through this evaluation, I demonstrate that using BEST-PTA leads to lower prediction error and fewer large errors.

4.2 Bayesian Estimator for Partial Trajectory Alignment (BEST-PTA)

In this section, I describe the various components of BEST-PTA, and discuss how these components fit into the overall partial alignment framework. Given a set of training trajectories, the first component of the system computes a spatially accurate representative trajectory, X^R, through an iterative process based on DTW. The next component then analyzes the velocity profile of this trajectory to discover stop segments. These segments and X^R are then fed into an optimization-based segmentation process that divides the representative trajectory such that overlapping trajectory regions are assigned to separate segments. Next, the framework learns a Random Forest classifier [59] over the computed segments, builds a library of segment-specific prior distributions over correspondence points, and learns a distance-based likelihood. During execution of a new trajectory, the framework combines the current segment

probabilities and durations from a state machine, the learned prior distributions, and the likelihood function to generate the distribution over correspondence points.

4.2.1 Representative Trajectory Computation

The first step of the framework is to define a representative trajectory, X^R, based on the training set, $\mathcal{D}^T$. In order to avoid smoothing out trajectory features by simply averaging over the trajectories in the set, I developed an iterative algorithm based on DTW. In this process, the algorithm first removes a trajectory at random from $\mathcal{D}^T$ and assigns it as the current estimate of the representative trajectory, denoted as $\hat{X}^R$. Then, at each successive iteration, the algorithm randomly selects another trajectory from $\mathcal{D}^T$ without replacement, computes an alignment to $\hat{X}^R$ with DTW, takes a weighted average of the aligned points (and their time stamps) with weights assigned in proportion to the number of trajectories that $\hat{X}^R$ is already composed of, and resamples the trajectory back to the sampling rate, τ. The resulting trajectory becomes the new $\hat{X}^R$, and the process repeats until all trajectories in $\mathcal{D}^T$ are incorporated. Figure 4-1 depicts an example of how a representative trajectory computed using our method maintains features that are lost when simply taking a mean over the trajectories in the training set.

4.2.2 Stop Segment Discovery

As discussed in Section 2.6.2, one aspect of motion trajectories that can cause alignment complications is the presence of stops within a given trajectory. This can be observed, for example, in the trajectories formed by people reaching for objects or walking to a location in order to perform a brief task before continuing their motion. Within a trajectory, a stop will result in a cluster of points, all within a very small distance of each other. As Euclidean distance is often used as an alignment metric, this can lead to errors in alignment. To address this problem, the second component of our framework incorporates a speed estimate in order to identify the

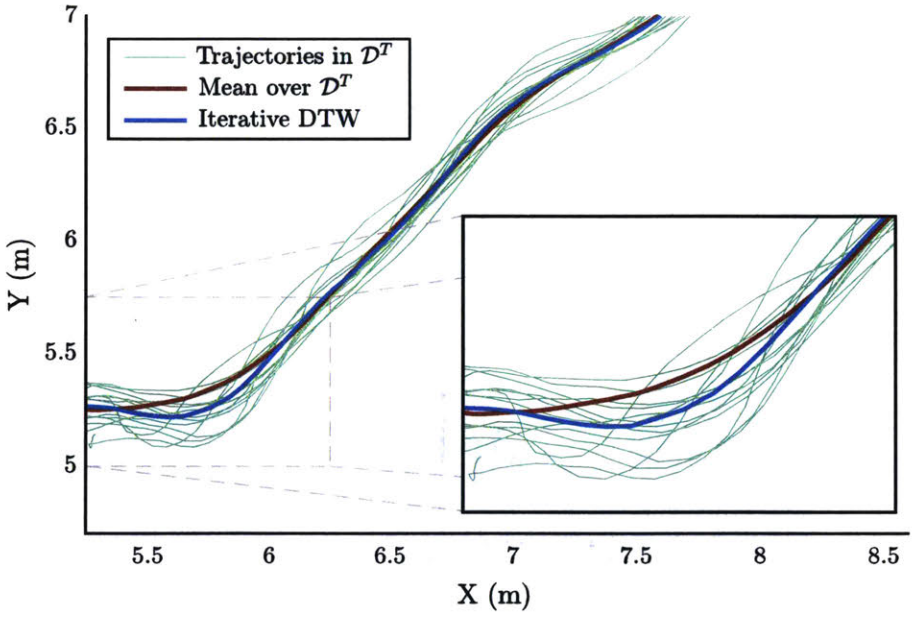


Figure 4-1: A comparison between computing the representative trajectory X^R by taking the mean of the trajectories in $\mathcal{D}^T$ and using our iterative DTW method. Note how the former "smoothes out" the corner and results in a representative trajectory that is spatially inconsistent.

dominant movement modes and finds potential "stop segments," which allows other components of our method to include special consideration for these unique regions.

The first step of stop segment discovery is to compute the speed at each point in the trajectory, denoted as $\vec{Z}$. This is performed by finding a cubic spline fit to the trajectory, taking a numerical derivative to determine velocities, and computing the norm of the velocity vector at each time step. The system then fits a Dirichlet process Gaussian mixture model (DPGMM) to these speeds, with the resulting mixture components representing the dominant speed modes within X^R. This model is particularly suitable because human motion characteristics tend to be Gaussian-distributed, and no assumptions must be made about the number of speed modes. Importantly, however, while movement modes will tend to be Gaussian-distributed, the stop mode will tend to be a half-Gaussian with a peak near zero. In order to create a full-Gaussian stop mode, the system utilizes an extended vector, $\vec{Z}^E = [-\vec{Z}, \vec{Z}]$, to which it fits the DPGMM. Figure 4-2 depicts a sample histogram of $\vec{Z}^E$ of a tra-

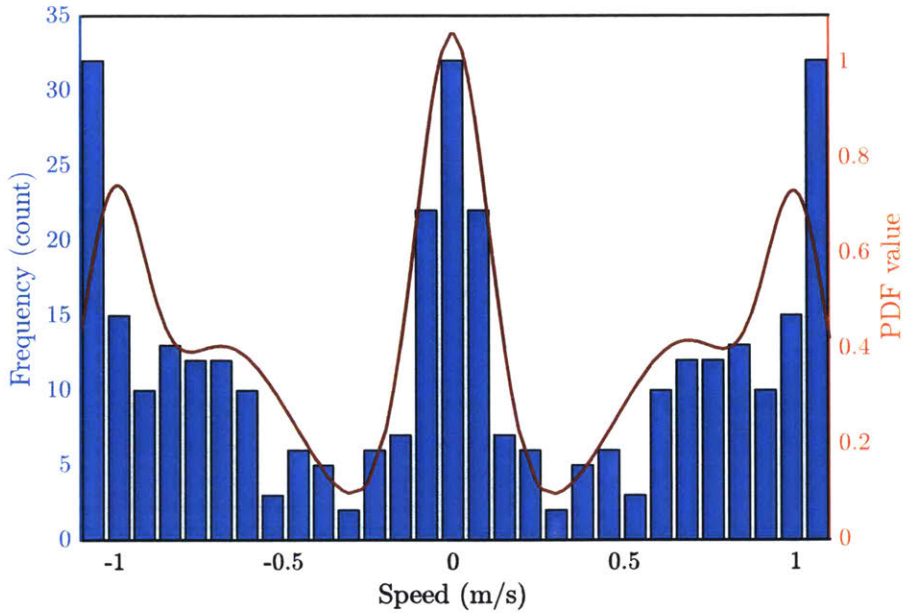


Figure 4-2: A histogram of $\vec{Z}^E$ and the DPGMM fit. The DPGMM shows a dominant speed mode at 0 (a "stop mode") and two movement modes at ± 0.7 m/s and ± 1.0 m/s.

jectory from the AA dataset and the corresponding DPGMM fit with a stop mode and two movement modes.

For every mixture component $i \in 1 \ldots n_c$ of the DPGMM, the system identifies it as a stop mode if it is centered around zero ($|\mu_i| < 1\frac{cm}{s}$) and its weight is at least 5% of what the value would be if the modes were uniformly distributed ($\pi_i > \frac{0.05}{n_c}$). Once a mixture component i is determined to be a stop mode, it can be used to define a speed threshold, $z^* = \mu_i + \sigma_i$; the stop segments are then defined as the portions of the trajectory below this threshold. To account for noise in the speed estimates, the algorithm adjusts stop and motion segments shorter than 7% of the trajectory length by merging them with neighboring segments. Furthermore, as the representative trajectory is derived from many individual trajectories, its speed profile can be significantly smoother than a typical trajectory for that action. To address this, the algorithm adjusts the speed threshold by using alignments to each of the trajectories in $\mathcal{D}^T$, taking the speeds observed within the stop segment regions, and fitting a new half-Gaussian to these speeds. The algorithm then recomputes the stop

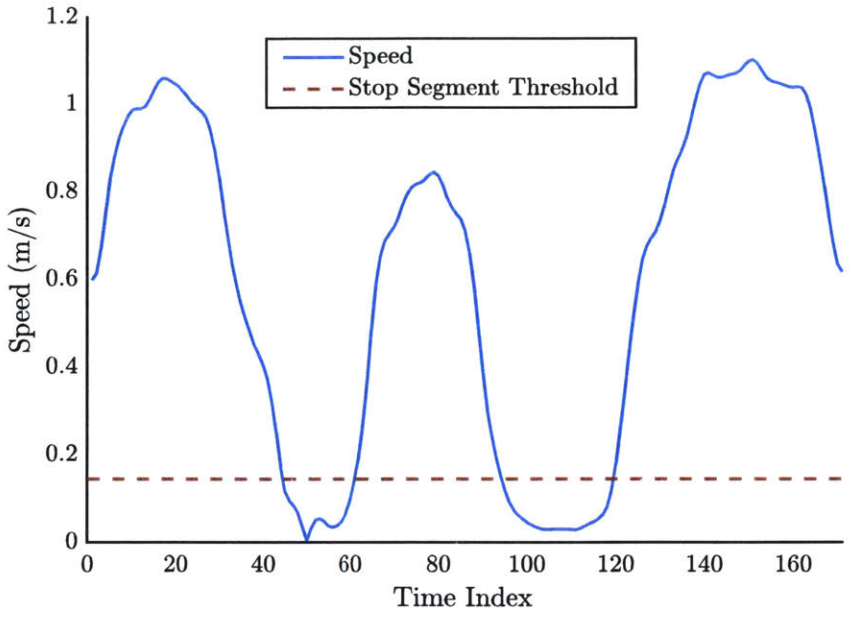


Figure 4-3: Speed vs. time for the same representative trajectory as the one used in Figure 4-2, showing the derived speed threshold, z^*.

segment bounds of X^R in the same manner as before, with a new z^* derived from the parameters of this new fit. These bounds are denoted as a set of start and end indices $\mathcal{W} = \{[w_1^s, w_1^e], \ldots, [w_{|\mathcal{W}|}^s, w_{|\mathcal{W}|}^e]\}$. Figure 4-3 depicts the speed profile of the same mean trajectory as that used to produce the histogram and DPGMM fit in Figure 4-2. In this figure, the dashed line represents the speed threshold derived from the mean and standard deviation of the discovered stop mode.

4.2.3 Ground Truth Alignment

As mentioned above, many partial trajectory alignment techniques rely upon the Euclidean norm as a distance metric between trajectories, which is problematic for trajectories that contain stop segments. This issue also arises in full trajectory alignment, which is necessary to compute ground truth alignments for the purpose of training and evaluating our approach. An example of a such a poor alignment, computed with DTW, is depicted in Figure 4-4. In order to generate more-accurate ground truth alignments between trajectories, I developed a novel algorithm for full

trajectory alignment that utilizes the stop segment detection component from the previous section.

Given a representative trajectory X^R and a target trajectory X^T, the first step is to find the boundaries of the stop segments within X^T using the speed threshold z^* and matching each discovered stop segment to those in X^R. The alignment within these stop segments is defined as a linear temporal spacing rounded to the nearest time step. The motion segments in X^R and X^T are then matched and aligned with DTW. As depicted in Figure 4-5, our method produces ground truth alignments that more accurately represent the relative progress through the trajectory at each point in time.

4.2.4 Trajectory Segmentation

Trajectories often contain spatially overlapping regions, which can also complicate partial trajectory alignment: similarly to the problem of temporary stops, overlapping regions can lead to small Euclidean distances between large portions of the trajectory, resulting in alignment errors. The objective of the third component of the BEST-PTA framework is to alleviate this problem by segmenting the representative trajectory X^R into m parts that include as little within-segment spatial overlap as possible. This is achieved by finding a vector of time indices, $\vec{S}$, that marks the boundaries of the segments (i.e., $\vec{S} = [1, s_2, \ldots, s_m, T]$).

I frame this segmentation as an optimization problem. A pair of points on a trajectory segment can be considered "overlapping" if the Euclidean distance between them is small but the difference between their respective time stamps is large. To capture this objective, the cost function in the range [0,1] for pairs of points at time steps i and j in a trajectory segment X is defined as follows:

$$C(i,j;X) = \left[1 - \tanh\left(\frac{3\|\vec{x}_i - \vec{x}_j\|^2}{\eta_D}\right)\right] \tanh\left(\frac{3|i-j|}{\eta_T}\right) \tag{4.1}$$

In this equation, η_D and η_T are normalizing factors for the distance and time differences, respectively. The former is set to the mean plus one standard deviation

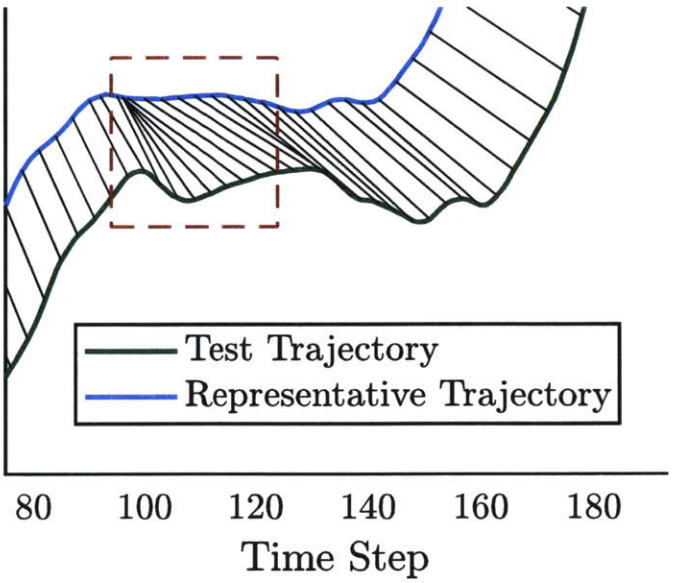


Figure 4-4: Part of a ground truth alignment between X^R and a test trajectory X^T found with DTW. Note the poor alignment within the stop region (dashed rectangle) when using DTW.

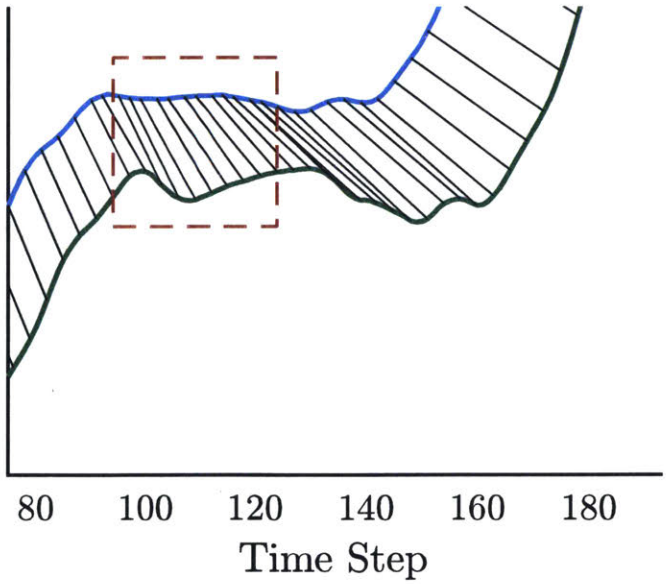


Figure 4-5: Part of a ground truth alignment between X^R and a test trajectory X^T found with our method.

of all pairwise distances in trajectory X, while the latter is set to $\frac{T}{2}$. The constant multiplications are included to shift the saturation of the hyperbolic tangent function to the point where the squared distance and time difference are equal to the respective normalizing factors. The mean cost of a segment with start and end indices t_s and t_e, respectively, is then defined as follows:

$$J(t_s, t_e; X) = \frac{1}{\binom{t_e - t_s}{2}} \sum_{i=t_s}^{t_e-1} \sum_{j=i+1}^{t_e} C(i, j; X) \tag{4.2}$$

In order to bias the optimization away from creating very short segments, a regularization term is introduced:

$$G(t_s, t_e; X, m) = 1 - \tanh\left(3m\frac{t_e - t_s}{T}\right) \tag{4.3}$$

Similarly to Equation 4.1, multiplication by a constant is performed so that the hyperbolic tangent function saturates, and thus brings the value of G near 0, when the length of this segment as a percentage of the total trajectory length equals $\frac{1}{m}$. Combining Equations 4.2 and 4.3, we arrive at the overall cost function for the segmentation:

$$R(\vec{S}; X) = \sum_{k=1}^{|\vec{S}|-1} J(s_k, s_{k+1}; X) + \lambda G(s_k, s_{k+1}; X, m) \tag{4.4}$$

In the above, λ controls the influence of the regularization term, which we set to 0.75 in our implementation. To identify the optimal segmentation, $\vec{S}^*$, it is necessary to identify the segmentation time index vector, $\vec{S}$, that minimizes Equation 4.4 — subject to the constraints that the first and last elements of $\vec{S}$ are 1 and T, respectively, that the elements of $\vec{S}$ are integers, and that they are monotonically increasing, as follows:

$$\begin{gathered} \vec{S}^* = \underset{\vec{S}}{\operatorname{argmin}}\, R(\vec{S}; X) \\ s_1 = 1; \quad s_{|\vec{S}|} = T; \quad s_i \in \mathbb{Z}; \quad s_i > s_{i-1} \end{gathered} \tag{4.5}$$

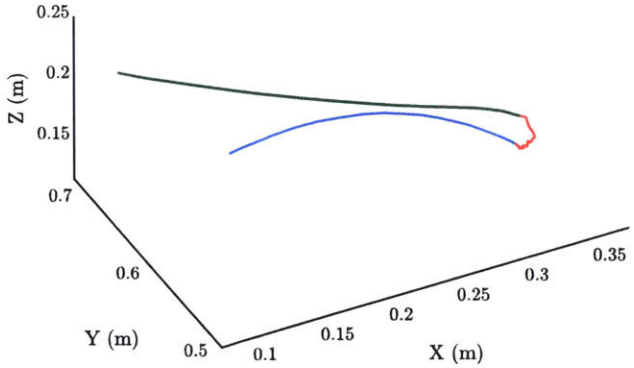
This formulation makes the segmentation problem an integer nonlinear program (INLP). As the overall objective function $R(\vec{S}; X)$ is trajectory-dependent and potentially non-convex, a global optimization method is necessary to calculate $\vec{S}^*$. In our implementation, we elected to use a genetic algorithm (GA) solver with a population size of 50, a crossover fraction of 0.8, and a constraint tolerance of 0.001.

If stop segments are identified as described in Section 4.2.2, the system incorporates them into the optimization as follows. First, the system enforces that the bounds of any identified stop segments in $\mathcal{W}$ are part of $\vec{S}$ and not allowed to change. Next, to prevent any of the free values of $\vec{S}$ from being placed between the start and end indices of any $\mathcal{W}_i$, we incorporated a nonlinear constraint that penalizes such placements proportional to the square of the encroachment:

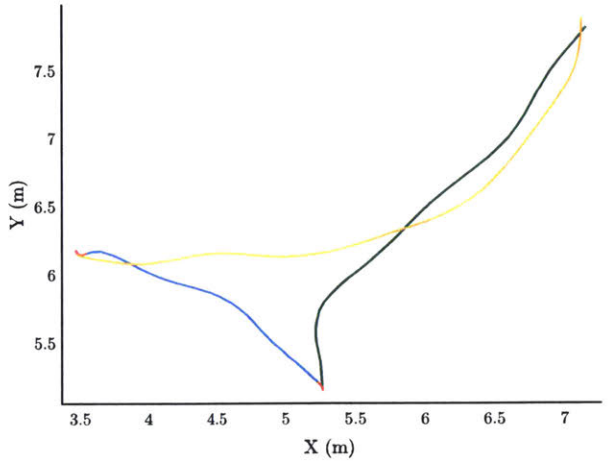
$$\sum_{i=1}^{|\mathcal{W}|} \sum_{j=1}^{|\vec{S}|} [w_j^s <= s_j <= w_i^e] \{min(s_j - w_i^s, w_i^e - s_j)\}^2 \leq 0 \tag{4.6}$$

Figure 4-6 shows examples of segmentations derived by using our optimization-based technique for sample trajectories from the TF, AA, and SSM datasets. As these trajectories all contain stops, the discovered stop segments fully define $\mathcal{W}$, and the optimization component is not needed. In order to show the resulting segmentations for trajectories containing only overlaps but not stop segments, the stop segments were first removed from the trajectories, resulting in segmentations depicted in Figure 4-7.

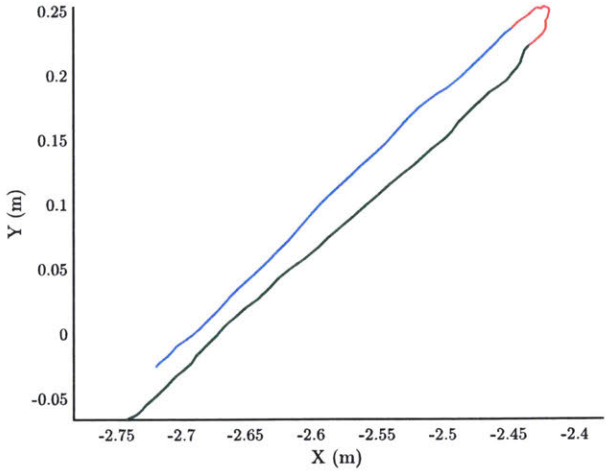
Based on the segmentations depicted in Figure 4-7, it might appear that using a maximum curvature segmentation could be sufficient, and that the presented optimization method is not necessary. One simple counter-example to this proposition is a circular motion with overlapping or near-overlapping regions. For example, a trajectory spiraling toward the middle has a gradually increasing curvature, with the highest value at the last point. A maximum curvature segmentation, therefore, would result in one long segment followed by short segments near the end of the trajectory, which fails at avoiding within-segment overlaps. This segmentation would be similar to the one depicted in Figure 4-8a. The optimization-based technique presented here, on the other hand, can easily provide a segmentation that results in no within-segment



(a) Representative trajectory from TF dataset.

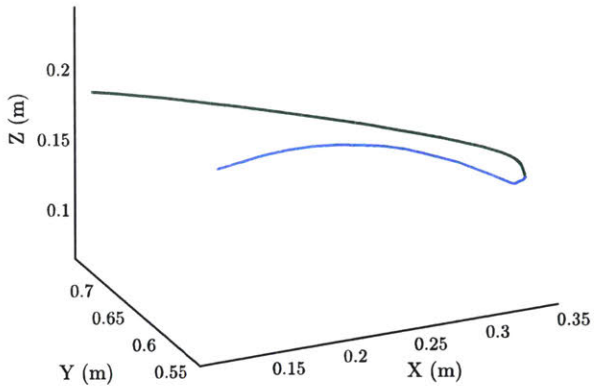


(b) Representative trajectory from AA dataset.

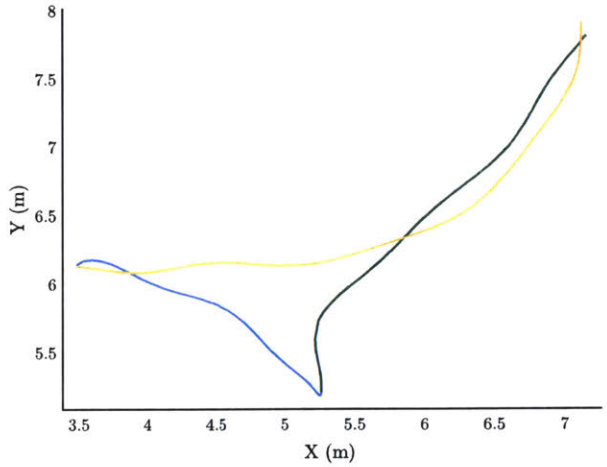


(c) Representative trajectory from SSM dataset.

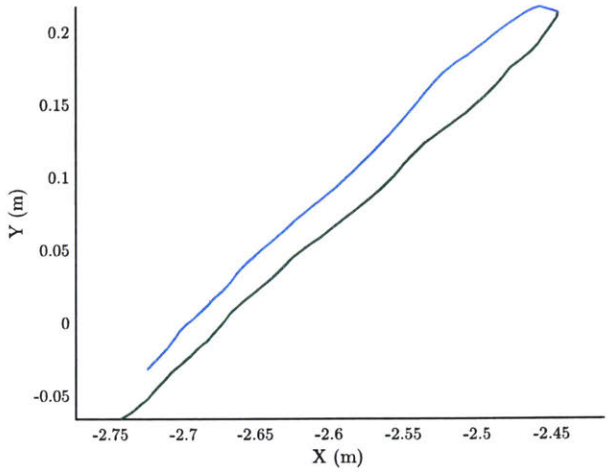
Figure 4-6: Example segmentations from each of the three human motion datasets. Each color defines a different segment, with red designating stop segments.



(a) Representative trajectory from TF dataset.

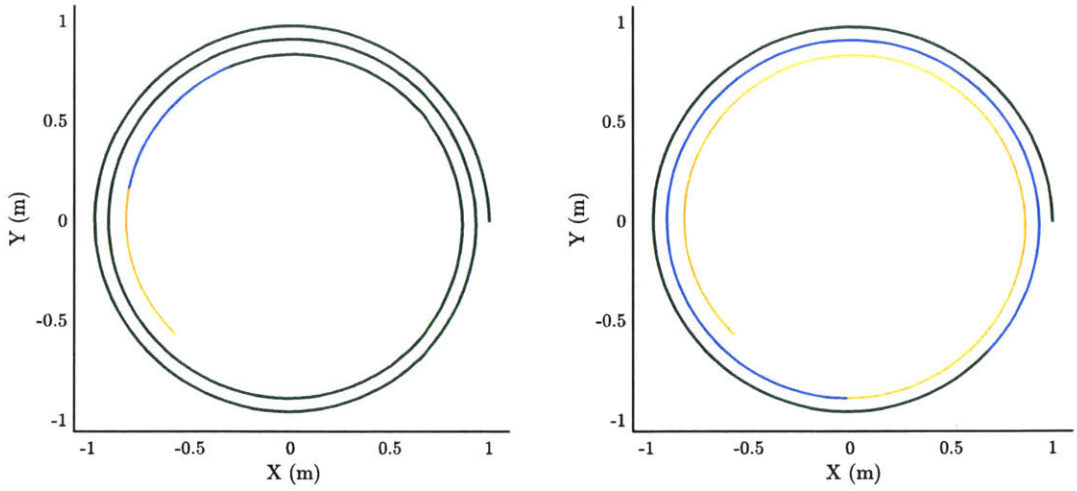


(b) Representative trajectory from AA dataset.



(c) Representative trajectory from SSM dataset.

Figure 4-7: Example segmentations from each of the three human motion datasets with the stop segments removed prior to segmentation. Each color defines a different segment.



(a) Maximum-curvature segmentation (b) Optimization-based segmentation

Figure 4-8: Segmentation of a spiraling trajectory with $m = 2$ using maximum-curvature segmentation (a) and the presented optimization-based method (b). The presented method succeeds in producing segments with minimal within-segment overlap, while a maximum-curvature segmentation does not.

overlaps, as shown in Figure 4-8b. The presented method, therefore, can achieve the goal of minimizing within-segment overlaps more robustly.

4.2.5 Segment Classification

Once the segmentation of the representative trajectory is complete, a classifier is required that would indicate which segment the head of the partial trajectory is currently in for each new partial trajectory observation $\vec{x}_t^P$. One common classification technique suitable for this task is the Random Forest (RF) [13], as it supports multi-class problems and its output is a discrete distribution over classes. In our case, the feature vector for a particular point $\vec{x}_t$ consists of a concatenation of trajectory coordinates, estimated velocities, and speed: $\vec{f_t} = [\vec{x}_t, \vec{v}_t, \|\vec{v}_t\|] \in \mathbb{R}^{2N+1}$ and the number of trees was set to 50.

To compute velocity estimates for the feature vector, the system removes high-frequency noise using the Savitzky-Golay differentiation filter [130]. The polynomial order and frame size for the Savitzky-Golay filter are hyperparameters that the system

learns through the use of a validation set, $\mathcal{D}^V$, based on the resulting classification accuracy. After training, given a feature vector corresponding to $\vec{x}_t^P$, the RF classifier outputs a discrete distribution over segments, denoted as $\theta \in \mathbb{R}^m$. This discrete distribution is then used by our segment state machine, which we describe in the following section.

4.2.6 Segment State Machine

In order to monitor which segment the head of the current partial trajectory occupies and how long ago each segment transition occurred, we implement a state machine. Denoting the indices of the segments as $k = 1 \ldots m$ and the vector of durations since transition to each segment as $\vec{D} = [d_1 \ldots d_m]$, once a state transition to segment k is detected, the state machine begins monitoring the duration of the current segment, d_k, while continuing to keep track of $d_1 \ldots d_{k-1}$.

For each new position of the current partial trajectory, $\vec{x}_t^P$, the state machine takes the current distribution over segments θ from the RF classifier and computes a moving average of these probabilities, denoted as $\hat{\theta} \in \mathbb{R}^m$, over a window of size γ_w.

The second parameter used in the state machine transition model is a temporal prior on segment durations. For each trajectory in the training set, ground truth alignments are computed, and the durations of each segment are collected. Next, the system fits a Gaussian Kernel distribution to the durations for each segment in order to characterize the prior probability of these durations. The cumulative distribution function of this distribution for segment k, denoted as $F(d_k)$, represents the probability that the segment is shorter than d_k, while the complement $1 - F(d_k)$ represents the probability that the segment is longer than d_k. Due to relatively small size of the three datasets, a uniform prior is added to the kernel distribution over the range of durations where $\epsilon < F(d_k) < 1 - \epsilon$ with $\epsilon = 0.001$. The uniform prior is defined as the product of the maximum value of the probability density function and a multiplier γ_u. The purpose of this prior is to prevent potential overfitting of this distribution to the small number of durations from the training set. Let us denote the

cumulative density function of the kernel distribution with the added uniform prior as $F^*(d_k)$.

Denoting the index of the current and next segments as c and n, respectively, during execution, as new points of the trajectory are received, the system determines whether a state transition to state n occurred via a Bayesian formulation using the segment likelihood and duration prior. Namely, based on the likelihood of each segment from the RF classifier, $\hat{\theta}$, and the prior probability of the current duration being shorter than d_c, $F^*(d_c)$, the state transition probability from state c to the next state is defined as:

$$P(c, n, d_c, \hat{\theta}) = F^*(d_c) \cdot \frac{\hat{\theta}_n}{\hat{\theta}_n + \hat{\theta}_c} \tag{4.7}$$

A state transition occurs when this state transition probability exceeds a parameter γ_n. Similarly, denoting the index of the previous segment as p, the probability of transitioning back one state from p is:

$$P(c, p, d_c, \hat{\theta}) = [1 - F^*(d_c)] \cdot \frac{\hat{\theta}_p}{\hat{\theta}_p + \hat{\theta}_c} \tag{4.8}$$

As for the forward state transition, the state machine transitions back to the previous segment when $P(c, p, d_c, \hat{\theta}) > \gamma_p$.

The hyperparameters $\vec{\Gamma} = [\gamma_w, \gamma_n, \gamma_p, \gamma_u]$ are learned with the use of the validation set, $\mathcal{D}^V$, by minimizing an error function based on the difference between the ground truth and estimated transition times and the number of back-transitions that occurred. Defining the ground truth transition time to segment $k > 1$ as t_k and a vector of estimated transition times as $\vec{t_k^E}$, with length defined by the number of back-transitions before the final estimate is made, the error function is the mean of the squared error between t_k and each of the values in $\vec{t_k^E}$ multiplied by a back-transition penalty $a^{|\vec{t_k^E}|-1}$. The constant a defines the extent of the penalty for multiple transitions, and is set to 1.1 in our implementation.

4.2.7 Temporal Correspondence Prior

The next component of the framework constructs the temporal prior of the Bayesian formulation. In order to make the prior robust to differences in encountered segment durations, we leverage the segmentation of X^R and build distributions based on the current segment k and durations sine transition to all previous states, $d_1 \dots d_k$, which are tracked by the state machine described in the previous section.

To build our prior distributions, the system iterates through all trajectories $X^i \in \mathcal{D}^T$ and computes ground truth alignments to X^R for each of them with the approach described in Section 4.2.3. As the system iterates through the training set, it counts, for each segment k and each time step in the range $1 \dots (s_{k+1} - s_k)$ within that segment, the number of times each of these time steps corresponded to each time step of the corresponding segment of X^R. This process results in a total of T vectors of such counts. The system then fits Gaussian kernel distributions to each of these vectors, generating smooth probability distributions without any assumptions about the underlying distribution structure. Each of these distributions, denoted as $P_{GK}(k, d_k)$, describes the prior probability of correspondence to the time steps of segment k for a given duration d_k since the transition to this segment occurred. In order to avoid poor Gaussian kernel fits to overly sparse data, these distributions are only defined for values of d_k for which there were at least 10 examples of in the training set. In other words, $P_{GK}(k, d_k)$ is only defined for durations up to the duration of the 10th longest example of segment k in $\mathcal{D}^T$. Let us denote this cutoff duration for segment k as d_k^C.

Next, we use the distributions $P_{GK}(k, d_k)$ to construct discrete priors over the time steps of X^R. First, we recognize that for any given segment k, all probabilities before time step s_k^* are zero. For the time steps of the segment, $s_k^* \leq t^* < s_{k+1}^*$, the system integrates each of the kernel distributions of segment k over intervals of length one centered around these time steps. Keeping in mind that $d_k = t^* - s_k^* + 1$, the discretized kernel distributions over possible values of t^* of X^R given a duration of d_k since transition to segment k and segmentation points $\vec{S}^*$ are then defined as follows:

$$P_{DKG}(t^*; k, d_k, \vec{S}^*) \propto \begin{cases} 0 & t^* < s_k^* \\ \int_{(t^*-s_k^*+1)-0.5}^{(t^*-s_k^*+1)+0.5} P_{GK}(k,t)dt & s_k^* \leq t^* < s_{k+1}^* \\ 0 & t^* \geq s_{k+1}^* \end{cases} \tag{4.9}$$

In order to account for durations longer than d_k^C, if such a duration is encountered, the distribution at d_k^C is taken and shifted forward in time by the difference between d_k^C and d_k. The final definition of the discrete, temporal prior probability over the time steps of X^R given a duration of d_k since transition to segment k and segmentation points $\vec{S}^*$ is then given as follows:

$$P_T(t^*; k, d_k, \vec{S}^*) \propto \begin{cases} P_{DKG}(t^*; k, d_k, \vec{S}^*) & d_k \leq d_k^C \\ P_{DKG}(t^* - (d_k - d_k^C); k, d_k^C, \vec{S}^*) & d_k > d_k^C \end{cases} \tag{4.10}$$

Similarly to the method used in the temporal priors of the state machine, in order to prevent potential overfitting caused by data sparsity, the system adds a uniform prior to the non-zero values of P_T. The magnitude of this prior is different for stop and motion segments, and is learned with the use of the validation set $\mathcal{D}^V$ by selecting magnitudes that result in the smallest mean alignment error.

During execution of a trajectory, the system utilizes the learned collection of P_T distributions to formulate our overall prior. The continualy incremented values of $\vec{D}$ of the segment state machine represent competing hypotheses of the true value of k and d_k (i.e., whether $\vec{x}_t^P$ is in segment k with duration d_k, or in some prior segment $k-j$ with duration d_{k-j}). In order to provide robust priors, especially near segment transitions, the system computes a weighted average over all P_T up to the current segment, indexed by c, proportional to the current estimate of segment probabilities, $\hat{\theta}$:

$$\widetilde{P}_T(t^*; \vec{D}, \hat{\theta}, \vec{S}^*) \propto \sum_{k=1}^{c} \hat{\theta}_k P_T(t^*; k, d_k, \vec{S}^*) \tag{4.11}$$

Finally, in order to bias the prior toward high-probability segments, we weigh the probability at each value of t^* proportionally to the probability of the segment that t^* belongs to. If we define the segment index of time step t in X^R as $k(t)$, then the final form of our temporal prior is:

$$P(t^*; \vec{D}, \hat{\theta}, \vec{S^*}) \propto \hat{\theta}_{k(t^*)} \widetilde{P}_T(t^*; \vec{D}, \hat{\theta}, \vec{S^*}) \tag{4.12}$$

This final formulation, given the candidate segment durations and probability estimates from the state machine, selects the appropriate learned priors and synthesizes them together in order to provide a discrete prior distribution over all possible values of t^* in X^R.

4.2.8 Distance-Based Likelihood

The second key probability distribution of the Bayesian formulation that the system learns during the training process is the likelihood of observing a specific position given the alignment time t^*. The system builds the likelihood model by characterizing the typical distance between aligned points within each of the m segments of X^R. Namely, the system finds the ground truth alignment to each trajectory in $\mathcal{D}^T$ and computes an exponential distribution fit for each segment (i.e., learns the rate parameter λ_k for $k = 1 \ldots m$) based on vectors of observed distances between pairs of aligned points within each segment. The likelihood of the position $\vec{x}$ given a proposed alignment point t^* is then simply given by:

$$P(\vec{x} \mid t^*; X^R, \vec{S^*}) \propto f(\|\vec{x} - \vec{x}^R_{t^*}\|; \lambda_{k(t^*)}) \tag{4.13}$$

In this equation, $f(x; \lambda)$ is the probability density function of an exponential distribution evaluated at x with rate parameter λ, and $k(t)$ is the segment index of time step t in X^R.

4.2.9 Bayesian Correspondence Distribution

Next, I describe the process of using the temporal prior and distance-based likelihood for computing the correspondence distribution over the time steps t^* of X^R given a partial trajectory X^P. As each successive point $\vec{x}_t^P$ is received, the system updates the segment state machine, computing new values for $\hat{\theta}$ and $\vec{D}$. Combining these values and Equations 4.12 and 4.13, we define the overall correspondence distribution using Bayes' rule:

$$P(t^* \mid \vec{x}_t^P; X^R, \vec{S}^*) \propto P(\vec{x}_t^P \mid t^*; X^R, \vec{S}^*) P(t^*; \vec{D}, \hat{\theta}, \vec{S}^*) \tag{4.14}$$

This distribution allows one not only to select a high-probability correspondence point t^*, but also to characterize the confidence in this alignment based on the relative probabilities of other candidate alignment points. Based on this formulation, the computational complexity of evaluating the state machine update, likelihood, and prior for each new point $\vec{x}_t^P$ all scale linearly with the length of X^R. Based on this fact, the overall complexity of BEST-PTA is $\mathcal{O}(n)$, which facilitates its use in online applications such as activity recognition or human motion prediction.

4.3 Evaluation of Partial Trajectory Alignment

The Bayesian approach to partial trajectory alignment presented in this chapter was specifically designed to accommodate human motion trajectories with stops and overlaps. In order to evaluate BEST-PTA's robustness to these qualities, the three human motion datasets introduced in Section 3.2 are used. On top of containing examples of both stops and overlaps, these datasets differ in many key motion, task, and sensing qualities, as was discussed in detail in Section 3.2.4. Consequently, these three datasets are suitable for assessing the robustness and accuracy of our method. In this section, I present an evaluation of BEST-PTA against three partial trajectory alignment baselines in order to demonstrate that my approach results in reduced alignment error.

4.3.1 Baseline Methods

For the evaluations, I compare BEST-PTA against three baseline methods: Online Dynamic Time Warping (O-DTW) [28], Optimal Sequence Bijection (OSB) [92], and the moving window (MW) approach from the original MPS implementation described in Chapter 3. We selected these baselines based on their capability for partial alignment using low-dimensional time series as input (e.g., trajectories instead of time series of images).

The first baseline method, O-DTW, introduced in Section 2.6.2 is an online variant of the very popular DTW algorithm. Unlike DTW, however, this method builds the warp path from the start of the trajectory and selects which candidate time indices of the warp matrix to explore based on the current most promising direction for alignment. O-DTW has two key parameters: the number of new warp matrix values to compute at each iteration (denoted as c) and the maximum number of times the search can move in the same direction of the warp matrix ($MaxRunCount$). We selected values for these parameters using the validation set $\mathcal{D}^V$ by performing a grid search over a set of discrete values.

The second baseline, OSB, was also introduced in Section 2.6.2. Specifically, the formulation based on work by Latecki et al [92] was utilized, but with modifications introduced by Koknar-Tezel [77]. One key attribute of this method that makes it a suitable choice for trajectory alignment is that it accommodates outliers in the data by allowing for the skipping of certain points in either the partial or full trajectories during alignment. The main parameters include the penalty for skipping elements of either trajectory (denoted as C) the maximum number of elements that can be skipped in a row, and a warping window size that limits how far off the diagonal of the warp matrix the search should be limited to. We set the first parameter, C, via our validation set $\mathcal{D}^V$, using the method suggested by the authors of [92]. For the remaining parameters, we used the authors' guidance, and assigned the same value to each of them. To select that value, we performed the same hyperparameter tuning method as that used for O-DTW.

One potential drawback of OSB is that it is designed specifically to search for potential alignments starting in any part of the full trajectory. As a result, it can sometimes identify a similar subsequence in the wrong section of the trajectory, especially for very short partial sequences. To avoid this problem (and make the comparison fair) we introduced an additional constraint to OSB that forced it to match the first elements of X^R and X^P to each other. Also, in order to generate a reference point for x_t^P, this latest point of X^P could not be skipped; consequently, we also introduced this as an additional constraint to OSB.

Finally, the MW baseline was introduced in the context of the MPS in Section 3.3.4. This technique uses a temporal moving window, centered on the current time step, to search for a point on X^R within this window that is closest to the current position. The only parameter was the size of the window over which to conduct the search for the nearest point in X^R. Specifically, we were looking for the number of time steps, w, such that the search was limited to $X^R_{[t-w,t+w]}$. Once again, we trained this parameter in the same manner used to train the other baselines.

4.3.2 Evaluation Method

A leave-one-out cross-validation (LOOCV) was performed on each of the three datasets in order to assess the alignment performance of BEST-PTA and the baselines. For each iteration of the cross-validation, one sequence of actions was withheld as the evaluation set $\mathcal{D}^E$ (i.e., a set with one of each type of action). The remainder of the data was randomly divided into a test set, $\mathcal{D}^T$, and a validation set, $\mathcal{D}^V$, with 70% assigned to the former and 30% assigned to the latter. As in the MPS evaluation in Section 3.4, the data was divided into these sets grouped by participant for the TF dataset, and randomly for the AA and SSM datasets.

The error metrics for this evaluation were defined in terms of the time difference (in seconds) between the proposed and ground truth partial trajectory alignment times t^* for each time step t of the evaluation trajectory. The ground truth alignment is computed as described in Section 4.2.3. The first metric considered is the mean alignment error for each dataset and method. The second metric involves the analy-

sis of the occurrence of large errors. As the use case of partial trajectory alignment considered in this thesis is human motion prediction, large errors in alignment can translate to large prediction errors, which can cause a robot's planner to generate unsafe or inefficient motions. Specifically, we consider the percentage of test trajectories for which large errors occurred for at least 5% of the trajectory. The 5% threshold was selected in order to distinguish between cases where there is a short spike in error for just a few time steps from those where there is a large and persistent error.

In order to assess the impact of the presence of stop segments on alignment performance, the analysis was performed on both the original trajectories (which contained stop segments) and modified trajectories with those segments removed for each action of each dataset. Below, we refer to these modified datasets by adding a "_NS" for "No Stop" (e.g., the version of the tabletop factory task with no stop segments is denoted as TF_NS).

4.3.3 Results and Discussion

Reduction in Alignment Error

The results of the evaluation indicate that BEST-PTA leads to superior alignment for both variations of all three datasets. The mean alignment errors, in seconds, for all of the datasets are summarized in Table 4.1. For the original TF dataset, BEST-PTA reduced the mean alignment error by 44.0%, 15.1%, 20.4% when compared to O-DTW, OSB, and MW, respectively. The reduction in error for the AA dataset was even more significant, with BEST-PTA outperforming the baselines by 69.4%, 55.6%, and 59.0%. Finally, for the SSM dataset, the errors were reduced by 73.4%, 37.6%, and 17.5%.

The results were similar for the modified datasets (where stop segments were removed) as well. BEST-PTA reduced the mean alignment error by 80.5%, 66.6%, and 46.6% for the TF_NS dataset, 85.5%, 68.6%, and 59.2% for the AA_NS dataset, and 85.5%, 68.6%, and 59.2% for the SSM_NS dataset when compared to O-DTW, OSB, and MW, respectively.

Dataset	O-DTW	OSB	MW	BEST-PTA	Friedman
TF	0.50±0.23	0.33±0.23	0.35±0.21	**0.28±0.21**	$\chi^2 = 108.68$
TF_NS	0.10±0.05	0.06±0.04	0.04±0.04	**0.02±0.02**	$\chi^2 = 339.06$
AA	0.28±0.13	0.19±0.15	0.21±0.14	**0.09±0.09**	$\chi^2 = 127.06$
AA_NS	0.15±0.07	0.07±0.05	0.05±0.06	**0.02±0.01**	$\chi^2 = 180.7$
SSM	1.23±0.68	0.53±0.38	0.40±0.34	**0.33±0.23**	$\chi^2 = 148.82$
SSM_NS	1.09±0.71	0.56±0.56	0.35±0.47	**0.21±0.13**	$\chi^2 = 149.98$

Table 4.1: Mean partial trajectory alignment errors in seconds ± standard deviation for the original and "No Stop" versions of the TF, AA, and SSM datasets. All Friedman tests and partwise comparisons with the Wilcoxon signed rank test are statistically significant ($p < 0.001$).

To assess statistical significance of these results, we applied the Friedman test to determine the main effect of the alignment method and the Wilcoxon signed rank test for pairwise comparisons. Both the Friedman test and the pairwise comparisons between BEST-PTA and the baselines were all statistically significant ($p < 0.001$ for all tests).

The mean alignment error results provide strong evidence for BEST-PTA being an effective partial trajectory alignment method for trajectories containing stop segments and overlapping regions. As BEST-PTA outperformed the baselines for all three datasets, these results also indicate that our approach is suitable for alignment of both ambulatory and manipulative motion trajectories, which amplifies its usability within human-robot interaction contexts.

Occurrence of Large Alignment Errors

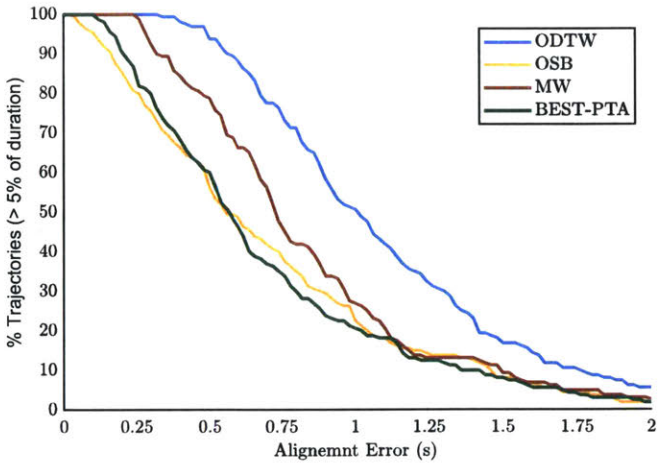
Noting the mean alignment error values in Table 4.1, while BEST-PTA outperformed the baseline methods, the differences in magnitude are on the order of tenths of seconds. The mean error, however, does not capture the entire benefit of our method, since large portions of the trajectories are simple to align, resulting in small mean errors. As mentioned above, however, while the overall mean error can be small, the

presence of large errors at key portions of the trajectory can lead to large errors when the alignment is used for human motion prediction.

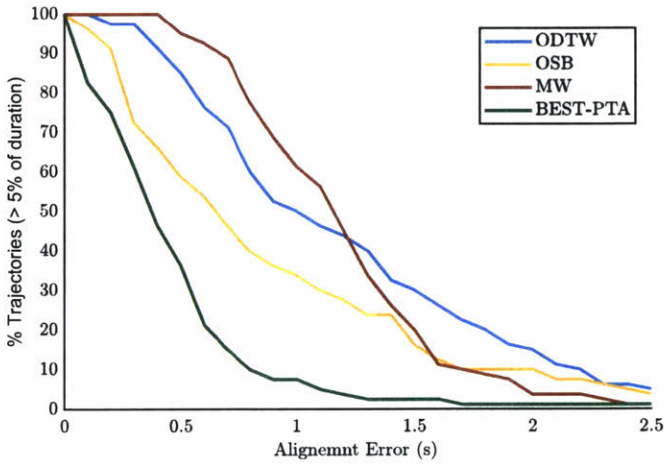
By utilizing a Bayesian formulation, a key benefit of our method is that it aids in reducing the occurrence of such large alignment errors. To gauge the effectiveness of our method in achieving this goal, an analysis characterizing the occurrence of large errors was performed. Namely for each test trajectory throughout the LOOCV, we assessed whether or not the alignment error exceeded various thresholds. In order to only consider trajectories where this error was persistent, a trajectory was considered to exceed a given error threshold if this threshold was surpassed for at least 5% of the trajectory. Figure 4-9 depicts the results of this analysis for the three standard datasets (with stop segments included).

The results from each of the datasets shown in this figure indicate that BEST-PTA was successful at keeping large errors from occurring often, either matching or outperforming the baseline alignment methods. The method was most successful in this aspect in the AA dataset (Figure 4-9b), in which it is apparent that BEST-PTA results in fewer instances of large errors. Note, for example, that alignment errors of at least 1s occurred in only 7.5% of the test trajectories when using BEST-PTA, while they occurred in 34%, 50%, and 61% of the trajectories when using the OSB, O-DTW, and MW approaches, respectively. Overall the occurrence of large errors dropped of very quickly as a function of the alignment error threshold; only 10% or fewer trajectories contain errors greater than 0.8s. The 10% bar is reached by the other methods at much higher alignment errors (1.7s, 1.9s, and 2.2s for the MW, OSB, and O-DTW methods, respectively).

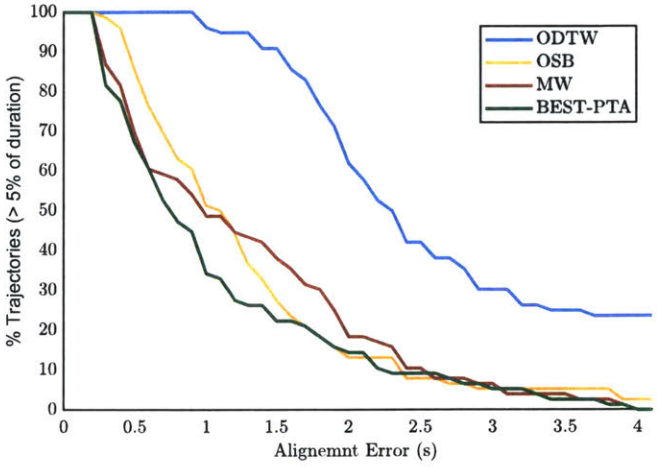
From the remaining subfigures in 4-9, one can see that BEST-PTA does not clearly outperform all of the baseline alignment methods in terms of reducing large errors in the TF (Figure 4-9a) and SSM (Figure 4-9c) datasets. While this is the case, the developed method still either matches or outperforms the baselines across all error thresholds. For example, while BEST-PTA performs nearly identically to OSB in the TF dataset, it outperforms the O-DTW method by a large margin throughout, and outperforms the MW method for errors of 1.2s or lower. In the SSM dataset, BEST-



(a) TF dataset



(b) AA dataset



(c) SSM dataset

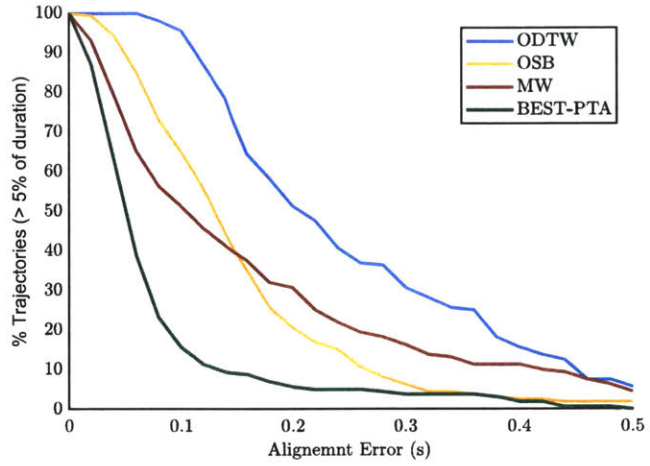
Figure 4-9: Percentage of trajectories in the three standard datasets for which the alignment error was above the given thresholds for at least 5% of the trajectory.

PTA once again very clearly outperforms O-DTW, and either matches or outperforms the other two baselines depending on the error threshold in question. Similar trends in relative performance can also be observed in Table 4.1.

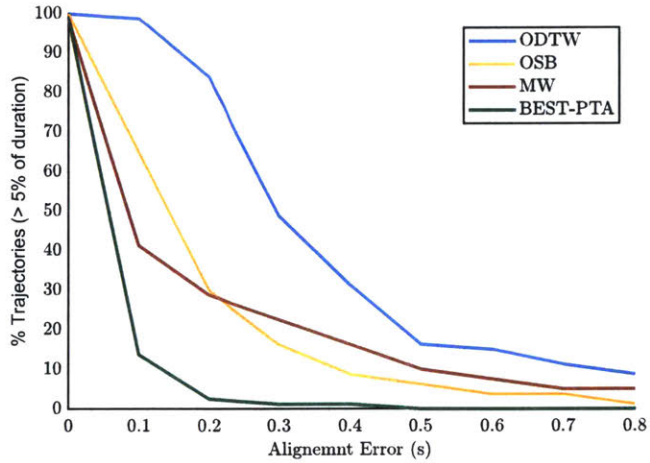
One explanation for why BEST-PTA does not manage to reduce large alignment errors as significantly in the TF and SSM datasets as it does in the AA dataset is the difference in stop segment durations. Namely, in these datasets the stops account for much larger portions of the trajectories, with stops representing a majority of the trajectory duration in the TF dataset and roughly a third for in the SSM datasets (see Table 3.1). While BEST-PTA attempts to specifically model progress through the stop segments based on durations observed in training, the temporal priors will guide the alignment in these regions toward a result based on the mean stop duration. In other words, each stop segment is expected to last as long as the mean stop segment duration observed in training, and until the segment classifier indicates the stop segment is over, this assumption will persist. As a result, if the stop segment is significantly shorter or longer than the mean duration, this will lead to large errors. Without any additional information, however, anticipating the length of the stop more accurately is not possible, making it difficult to remove these large errors.

The proposition that the large errors in the TF and SSM dataset are due to the presence of long stop segments is supported by the results obtained from the dataset variants in which stop segments were removed. In these datasets, our method is much more effective at reducing the number of large errors when compared with the baselines, as can be seen in Figure 4-10.

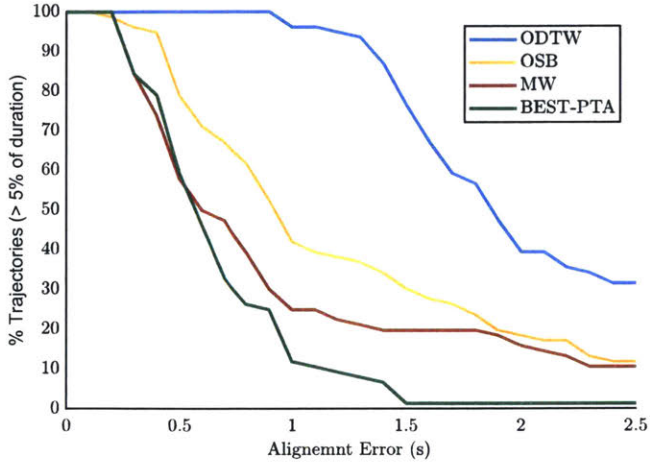
Overall, the fact that our approach leads to either similar numbers or significantly fewer large errors has potentially significant implications for the application of partial trajectory alignment to human-robot interaction applications such as activity recognition or motion prediction. In the case of human motion prediction, for example, reducing the number of large alignment errors is likely to result in fewer instances of significant prediction errors. This is the topic of the second evaluation described in Section 4.4.



(a) TF_NS dataset



(b) AA_NS dataset



(c) SSM_NS dataset

Figure 4-10: Percentage of trajectories in the three datasets with stop segments removed for which the alignment error was above the given thresholds for at least 5% of the trajectory.

4.4 Evaluation of Impact on Human Motion Prediction

As the overall goal of this thesis is to develop methods for supporting robust and accurate human motion prediction, an assessment of the effect of BEST-PTA on prediction performance was carried out. Based on the fact that the first evaluation showed that the developed method reduced mean alignment error and the occurrence of large errors, how this better alignment affects the accuracy of prediction is of interest.

4.4.1 Evaluation Strategy

In order to assess the effects of using BEST-PTA on the quality of human motion prediction, a LOOCV was carried out using the same data splits as in the evaluation of alignment error. The MPS was used as the method of prediction, with the same set of predictors as those described in Section 3.3.4. The training and validation sets were used as the $\mathcal{D}_{Train}$ and $\mathcal{D}_{ModelSelection}$ datasets from the MPS formulation.

For each of the three baselines and BEST-PTA, the pre-trained aligners from the first evaluation were set as the alignment method of the MPS, and the rest of the training process was then carried out as described in Section 3.3. The trained MPS was then used to make predictions on the evaluation set, $\mathcal{D}^E$. Similarly to the first evaluation, both mean prediction errors and the occurrence of large errors were analyzed.

4.4.2 Results and Discussion

The results of the evaluation support the hypothesis that utilizing BEST-PTA for partial trajectory alignment leads to improved human motion prediction. Overall, the mean prediction errors of the MPS for the TF dataset were reduced by 12.8%, 9.9%, and 8.8% when compared to the results when using O-DTW, OSB, and MW, respectively. For the AA dataset, the reduction in error was even more significant,

with BEST-PTA resulting in a reducing in error by 28.0%, 17.7%, and 19.9% when compared to the baselines. Finally, for the SSM dataset, the reduction in error was 22.8% and 5.1% for the O-DTW and OSB methods, respectively, but there was no significant change between BEST-PTA an MW. Once again, a Friedman test and pairwise comparisons with the Wilcoxon signed ranks test showed these results to be statistically significant ($p < 0.001$ for all tests with the exception of the pairwise test between BEST-PTA and MW mentioned above).

Due to BEST-PTA's quality of having fewer large alignment errors, large prediction errors were also reduced when it was used with the MPS. As mentioned above, BEST-PTA led to most significant reductions in large errors in the AA dataset. Figure 4-11 shows that the reduction in alignment indeed results in reduced prediction error as well. Similarly to before, the figure depicts the percentage of trajectories in the AA dataset for which the error in human motion prediction was greater than the given thresholds for at least 5% of the trajectory for three prediction time horizons.

From this figure, we can see that BEST-PTA indeed allows the MPS to make fewer large errors. For example, at a time horizon of 3.7s, an error of 1.5m was found in 15% of the trajectories, while this same error was found in 30% of trajectories when using the next-best aligner. As the workspace in the AA dataset was 3.5m by 10m and typical one-way motions were on the order of 5m, an error of 1.5m is substantial, and so reducing its occurrence can have a positive impact on the quality of the interaction. If a robot's planner is using these predictions to make decisions about its own motions in a shared human-robot workspace, large prediction errors can cause the robot to take inefficient or unsafe actions. Consequently, utilizing BEST-PTA as part of a prediction framework can lead to improvements in safety and efficiency of human-robot interaction.

4.5 Conclusions and Future Work

In this chapter, I introduced BEST-PTA, a Bayesian estimator for partial trajectory alignment. This method is specifically designed to accommodate the presence of tem-

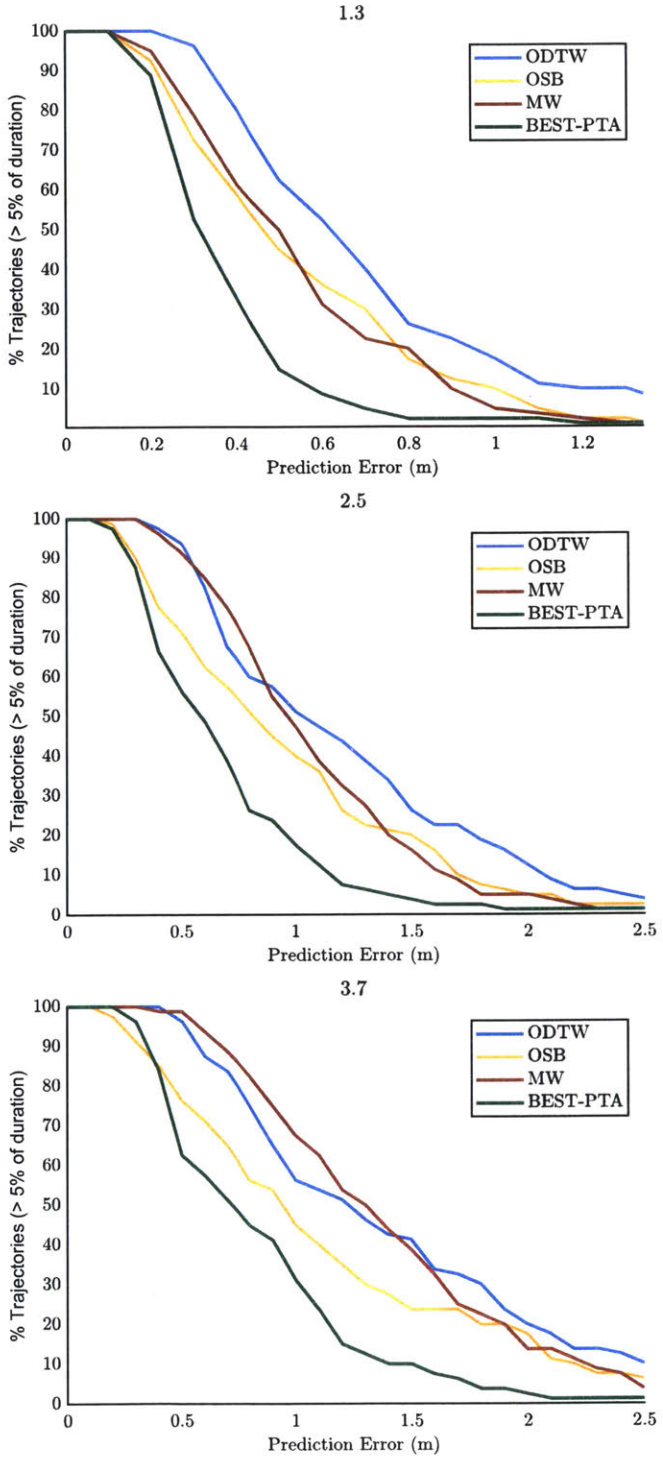


Figure 4-11: Percentage of trajectories in the AA dataset for which the MPS prediction error was above the given thresholds for at least 5% of the trajectory for time horizons of 1.3s, 2.5s, and 3.7s.

porary stops and overlapping trajectory segments. Due to its linear time complexity and the ability to provide distributions over candidate alignment points, my method is suitable for online applications such as activity recognition, action and motion prediction, and gesture recognition. I evaluated this technique against state-of-the-art baselines, and demonstrated that BEST-PTA outperforms these baselines for both ambulatory and manipulative human motions. Furthermore, I showed the utility of our method for human motion prediction by integrating it as part of a previously developed prediction framework, and demonstrated that it leads to lower prediction errors.

One limitation of the developed method is the rigid structure of the segment state machine component of the framework. While the current Bayesian state transition method attempts to balance a duration prior with segment likelihoods, deviation from typical durations or motions can lead to late or early state transitions. A probabilistic method of representing likely state and duration pairs could be implemented in order to introduce a more flexible way of tracking these values.

Another noteworthy drawback of the developed method is the limited resilience to anomalous trajectories. While the Bayesian estimation of alignment allows for some flexibility, trajectories that differ significantly from those observed in training can lead to large alignment errors. An unexpected extra stop, for example, can confuse the segment classifier, leading to an unwarranted early state transition. Detection of anomalous trajectories could allow for more robustness by, for example, falling back on a "single segment" solution in which the temporal prior is based on the entire trajectory instead of individual segments.

Finally, the segmentation component of the framework could be extended to use a nonparametric approach to finding the best number of segments in a given representative trajectory. While the appropriate number of segments can sometimes be apparent based on the trajectory, this is not necessarily true in general.

Chapter 5

Evaluation of Safety and Efficiency of Human-Robot Interaction

In the previous chapters, I described methods for improving the robustness and accuracy of human motion prediction. As the objective of this thesis is to use such methods in order to support safe and efficient human-robot interaction, in this chapter I present various evaluations of these methods in order to assess their impact on HRI and provide proof-of-concept implementations [1].

The HRI scenario used in this part of the thesis is the co-navigation in an automotive assembly environment (the AA dataset). The domain of automotive assembly has several near-term use cases of human-robot interaction under investigation by researchers [1, 132]. These include stationary robots designed to provide ergonomic benefits to human associates during assembly tasks, such as cockpit installation [7], rear axle assembly [131], or door sealant application [2]. A few robots with varied degree of mobility have also been researched and developed [4, 54, 148, 123], including "Robot Workmate" a system with single-axis mobility for inspection tasks [107].

In order to provide a proof-of-concept implementation demonstrating the benefits of the developed human motion prediction methods, first we developed an integrated prediction and planning framework. The system employs human motion prediction in conjunction with a complete, time-optimal path planner to execute motions in

[1] The content presented in this chapter is part of a collaborative project published in [147].

the shared environment. For human motion prediction, collaboration with a mobile robot in the manufacturing domain requires accurate predictions over both short and long time horizons. For example, the robot must know where a person will be in the short term in order to maintain effective collision avoidance, but also must know the human's long-term predicted path in order to plan efficient motion toward its own goal. We use the Multiple-Predictor System for prediction in order to assess the capability of the in providing accurate prediction for a range of time horizons.

The robot also requires a method for adapting its own behavior based on the knowledge of human behavior provided by the MPS. For example, if the robot receives predictions indicating that a human will cross in front of it, the robot can plan to either yield the way to the human or continue moving depending upon when the cross is predicted to occur. However, the system has limited freedom to perform such adaptations due to its single-axis mobility. This necessitates an approach that can generate plans quickly while reasoning about time and predictions. Schedule considerations drive the production environment and it is also crucial that the online planning system incorporates an explicit representation of time (i.e. performs planning in time). Thus, we use the Safe-Interval Path Planner (SIPP), a time-optimal search algorithm for planning in time, to plan robot trajectories [118].

SIPP generates plans under the assumption that the available predictions are fixed and accurate; however, in practice, predictions evolve as available information changes during task execution. The physical position of the robot will also change during the time-critical planning process. Hence, along with SIPP, we incorporate an algorithm to interleave prediction and planning with the execution of robot motion.

Next, I will describe an implementation of this integrated prediction and planning framework on a real physical system. The objective of this implementation was to qualitatively assess how the developed integrated framework affects human-robot interaction and to provide a proof-of-concept implementation in support of online use of the MPS. Through this physical demonstration, I demonstrate that the MPS is capable of providing predictions in an online manner. Furthermore, I show that

using the MPS as part of the developed prediction and planning framework leads to adaptive robot behavior and improved interaction.

Finally, in order to provide a more quantitative assessment of the developed prediction and planning framework, I describe a set of simulated evaluations. Namely, the objective of these evaluations is to assess improvements to safety and efficiency, compared to state-of-the art approaches applicable to factory environments. Results demonstrated reductions in safety-related stops, decreases in task times, and improvements in measures of fluency of interaction.

5.1 Integrating Prediction with Planning in Time

The first step in evaluating the feasibility and benefit of using the human motion prediction approaches developed in this thesis involves developing an integrated prediction and planning framework. This framework needs to utilize the predictions at various time horizons provided by the MPS in order to plan a safe and efficient robot motion. Furthermore, the framework needs to be computationally efficient in order to operate in an online manner. In this section, I describe such a prediction and planning framework, composed of human motion prediction with the MPS and robot motion planning in time.

5.1.1 Prediction with the MPS

For the prediction component of the framework, I utilize the Multiple-Predictor System, introduced in Chapter 3. The implemented version of the MPS for these evaluations is composed of the same three prediction methods as those described in that chapter. The first method, velocity-based position projection (VBPP), estimates future locations by projecting the human's current position through an estimate of his or her velocity as computed via the Savitzky-Golay Filter [130]. Once the velocity is estimated, the VBPP assumes that the person will continue moving at that speed for the duration of the time horizon of interest. The second predictor is the time series classification (TSC) method, which builds upon the goal-based time series classifica-

tion approach presented by Pérez-D'Arpino and Shah [117]. The final predictor in the MPS is a sequence prediction (SP) method that reasons on observed sequences of actions instead of the motion itself. The sequence prediction method uses previously observed action sequences to learn which sets of actions occur before others [93].

5.1.2 Planning Robot Motions in Time

Next, in order to utilize the predictions at various time horizons provided by the MPS, we require a suitable robot motion planner. Planning with predictions, however, can be computationally expensive. State-of-the-art human-aware collaborative manipulation systems typically utilize short-term predictions [116]; however, predictions involving significantly longer time horizons may be useful for planning long robot paths ($\approx$ 10m) in a factory setting. Further, in the Automotive Assembly (AA) scenario used in this chapter, the robot is constrained to move along a rail (see Section 3.2). As a result, our robotic system has limited freedom to adapt to the behavior of nearby humans. If the planning time is lengthy and the robot reacts to human too late, it might be unable to exhibit anticipatory behavior despite the existence of predictions that include the desired anticipatory information. This emphasizes the need for computationally tractable approaches that account for the critical effect of timing for path planning, and allow for interleaving online planning with execution.

Safe-Interval Path Planning

To account for the time-critical nature of the robot planning problem, we use a representation that explicitly models time. We incorporate SIPP as the underlying planner for our system [118]. By using safe time-intervals, SIPP significantly reduces the cardinality of the state space and provides a computationally efficient approach for planning despite explicit modeling of time. Using time-indexed predictions of human motion, SIPP provides a feasible robot trajectory if one exists, and returns a failure otherwise. Given accurate predictions, SIPP is both complete and time-optimal. We

Algorithm 1 Interleaving Planning and Execution

```
1: procedure MAIN(R_goal)
2:     H_predictions ← MPS()                                  ▷ Get predictions
3:     R_start ← R_position                                   ▷ Get robot start state
4:     R_trajectory ← SIPP(R_start, R_goal, H_predictions)    ▷ Plan robot trajectory
5:     while Goal not reached do                              ▷ Executed at replanning rate
6:         if Safety stop triggered then
7:             return False
8:         Update robot command (R_command) using R_trajectory
9:         Issue R_command to hardware controller
10:        H_predictions ← MPS()                              ▷ Update predictions
11:        replan ← Predictions changed or R_trajectory is empty
12:        if replan then
13:            R_start ← R_command                            ▷ Update start state
14:            R_new-trajectory ← SIPP(R_start, R_goal, H_predictions)   ▷ Replan
15:            if PLANCHANGED (R_trajectory, R_new-trajectory) then
16:                R_trajectory ← R_new-trajectory
17:                Issue stop R_command to hardware controller
18:    return True
```

implement the planner as an extension to the Search-Based Planning Library [5] and ROS.

Interleaving Planning and Execution

Due to changing human motion predictions, SIPP alone is not sufficient for executing robot motion. SIPP assumes the predictions to be accurate, and thus does not include a mechanism for updating the robot plan online. To incorporate the latest predictions, our robotic system additionally requires online replanning, during which the robot itself may be in motion. It is undesirable to stop the robot during replanning; therefore, an approach to interleave replanning and execution is required.

We provide Algorithm 1 in order to achieve interleaved planning and execution. Upon receiving the goal location as input, Algorithm 1 uses the current predictions from the MPS, and the start state of the robot, to compute the robot's trajectory via SIPP (lines 1-4). To interleave trajectory execution and planning, motion commands are executed using the planned trajectory, which is updated at the replanning rate (lines 5-17). The algorithm returns a failure if a safety stop is triggered during execution (lines 6-7); to complete the goal once the stop is deactivated, the goal is reissued in order to reinitiate Algorithm 1.

As long as the goal is not reached and a safety stop is not triggered, motion commands are sent to the hardware controller according to the latest planned trajectory (lines 8-9). If SIPP could not identify a feasible solution a stop command is issued to the robot. The robot then updates its safe time-interval representation according to the latest available predictions (line 10). If the predictions change, or if the previous planner call returned a failure, replanning is performed (line 12-17). To account for change to the robot state (due to motion) between when the planner is called and the resulting plan is executed, the latest commanded pose of the robot is used as the start state for replanning with SIPP.

Human motion predictions continue to evolve over the course of task execution; however, not all changes to predictions result in plan changes. The new plan, generated in line 14, is thus compared against the plan currently being executed via the PLANCHANGED() method. The method returns success if either the trajectory lengths differ by a time threshold (1s) or the L^∞-norm of the difference in the two trajectories exceeds a distance threshold (0.5m), and prevents fluctuations in robot trajectory due to minor updates to predictions. If PLANCHANGED() returns success, the robot is stopped and the updated trajectory is used for execution from the following timestep. The algorithm returns success once the goal has been reached (line 17). By interleaving planning in time and motion execution, our system can adapt quickly and efficiently to the behavior of nearby humans.

5.2 Demonstration on Physical System

In this section, I describe an implementation of the above prediction and planning framework on a physical robotic system in an emulated automotive assembly work cell. The task in this physical demonstration is similar to the one presented in the description of the AA dataset in Section 3.2, and is depicted in Figure 5-1.

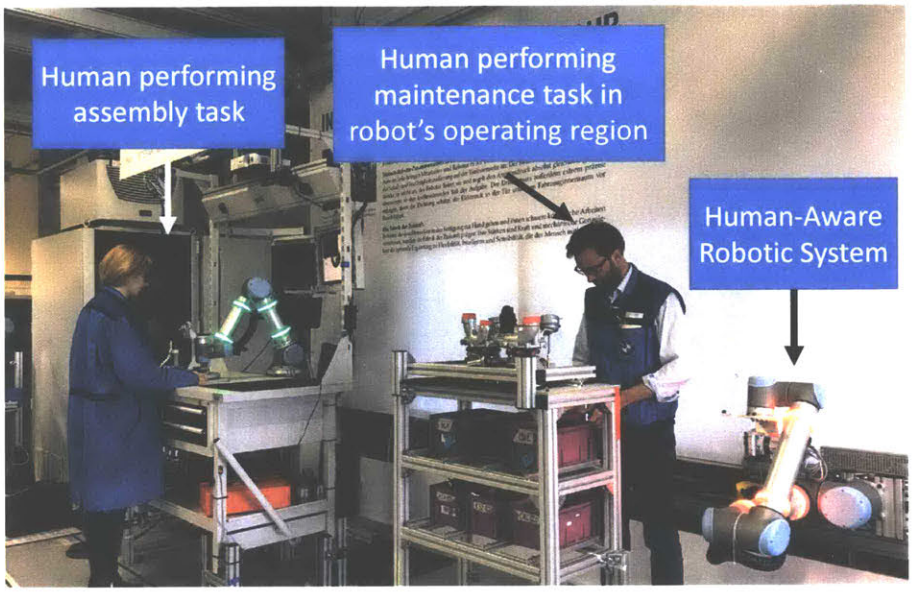


Figure 5-1: The physical work cell and emulated factory task used in the proof-of-concept prediction and planning framework. The human-aware robotic system delivering parts to a human associate during an engine assembly task, while another human performing an unrelated task intercepts its path between the depot and the workstation.

5.2.1 Task Description

The primary task of the system is to transit between workstations and part depots to deliver parts to human associates. These humans remain at their workstations throughout the task and use the parts delivered by the robot to assemble the engine. As engine assembly is a repetitive task, the parts and the order in which they are to be delivered remains fixed – i.e., the robot's task plan (or, equivalently, the sequence of goal locations) is pre-specified.

The set of possible human motions within the work cell is depicted in Fig. 5-2. The humans moving along these paths differ from the one at the workstation, and include workers responsible for stocking part depots and cleaning. While the sequence of robot goal locations is predetermined, it requires algorithms to plan and execute its trajectory to each goal. Humans within the shared environment may exhibit arbitrary motions (from the known set of possible motions) at any time during the robot's operation.

Figure 5-2: Work cell used in our physical demonstration. The three green arrows depict the possible actions the human could take in the workspace. The robot can move anywhere along the rail on the right. The workspace is the station directly next to the person, while the parts depot is the metal structure with purple-colored boxes.

5.2.2 Human Motion Data for MPS

In order to deploy the MPS in the physical work cell, a method of human motion tracking was required. The two-dimensional position of the person was tracked with the use of a Microsoft Kinect and the OpenNI framework. The data was collected at a rate of 10Hz. Figure 5-2 depicts the workspace and the three possible human actions shown as green arrows. The upper left corner of the figure is the view from the Kinect camera. In this task each action was a movement from the workspace (the location of the human in Figure 5-2), to either of the two long sides of the parts depot or a workstation on the side, a short waiting period at the depot or workstation, and then a return to the workstation. The robot was stationary during data acquisition, stopped behind the workstation.

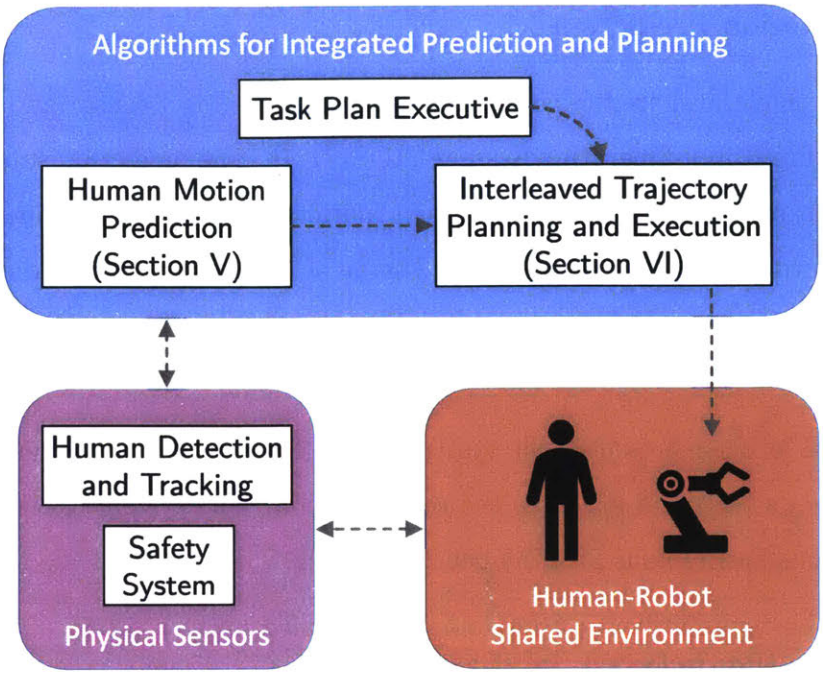


Figure 5-3: An overview of the human-aware robotic system.

5.2.3 System Overview

In this section, I briefly summarize the various components of the human-aware robotic system (see Figure 5-3).

Physical Robot

A UR10 collaborative robot serves as the robotic arm for performing manipulation tasks. Desired system mobility is achieved by mounting the arm on a linear axis unit. Arm joint angles are controlled using Universal Robot on-board controllers, and the arm is held in a fixed configuration while in motion due to the linear axis unit.

Safety System

A 2-D laser scanner is mounted on the robot; the scanner triggers a safety stop when any human (or object) is within the safety radius of the robot. A threaded implementation is used for the safety system. Once the stop is triggered, the entire physical robotic system is rendered immobile until the human leaves the safety radius.

Human Motion Prediction

Humans within the shared region are tracked using a Kinect sensor and the OpenNI tracker within Robot Operating System (ROS) [121]. The prediction sub-system uses the MPS to provide predictions of human motion, updated at a frequency of 5 Hz. The 2D coordinates of the human's head serve as features for the predictors.

Trajectory Planning and Execution

Once a goal location is issued, the system described in Section 5.1.2 generates a plan for the robot's base and executes the robot motion. The system checks for changes in predictions and replans at a frequency of 10 Hz.

Communication between Sub-Systems

Communication between the physical robot components (UR10, linear axis unit and the sensors) and the software is implemented using a programmable logic controller, TCP/IP sockets and ROS. A task plan executive issues a predetermined sequence of goals, based on the task plan, to initiate the robot task.

5.2.4 Demonstrations of Physical System Implementation

We demonstrated our system using the physical robots described in Section 5.2.3 within the environment depicted in Fig. 5-1 in a BMW test environment. In our demonstrations, the robot operated in one of three modes. In all modes, the current task and goal of the human was unknown to the robot. In the "Baseline" mode, no information about the human's current or predicted position is given to the planner, and the robot simply pauses its motion whenever a human enters the shared area of the work cell. This mode emulates the behavior of state-of-the-art reactive safety systems designed for factory environments, such as SafetyEye [3]. In the "Planning with Detection" mode, the planner incorporates the human's current position by assuming that the human will remain in that location until a new position is received. Finally, in our approach (the "Planning with Prediction" mode), the robot uses both

the currently detected human location and a set of position predictions obtained from the MPS. These predictions are made at a discrete set of time horizons ranging from 0.1-3 s in increments of 0.1 s, and are recomputed at a rate of 5 Hz. Video attachments representing the operation of these three modes are included with the paper and also available at `http://tiny.cc/cobotSAM` (see Fig. 5-4 for illustrative snapshots).

Through the results from our integrated system demonstrations, we observed that our system can anticipate and adapt to the behavior of nearby humans when utilizing both prediction and planning techniques. In one example of an observed adaptive behavior, the robot paused its motion and even moved backwards (Fig. 5-4(d)) to allow a human to reach the depot, then automatically resumed its task after the person moved back toward the workbench. Importantly, this behavior was automatically derived during execution, without the need for preprogramming. In contrast, when planning without anticipation of human motion, the robot was unable to effectively adapt its motion to unexpected human behavior, resulting in less fluent interaction, with the robot blocking the human's path and triggering safety stop more frequently (Fig. 5-4(c)). Lastly, while in the Baseline mode, the system yielded safe but inefficient robot motion (Fig. 5-4(b)), as the robot often stopped unnecessarily, sometimes in a position that interfered with the human's path.

5.3 Quantitative Assessment of Integrated System in Simulation

While the physical demonstrations provide intuitive, qualitative examples of the benefits of our system, the context for these demonstrations consisted of one robot assisting a single worker in a small work cell, which is not representative of how a robotic assistant would be deployed in a real factory. Therefore, we also performed a more thorough, quantitative analysis of the benefits of our system in simulation, with a much larger work cell in the analogue factory environment of the AA dataset presented in Section 3.2 and depicted in Figure 3-2.

In the simulation, the robot performed a pre-set sequence of tasks, simulating the pickup and delivery of components between depots and workbenches. The robot utilized the planning approach described in Section 5.1.2 to plan its motions toward the task plan's goal locations. We used a grid size of 10 cm and a replanning rate of 10 Hz for the planner, and set the maximum speed of the robot to 1 m/s. We simulated the laser scanner safety system by stopping the robot whenever a human came within a safety radius of 0.75 m. The simulated robot operated in the same three modes as those used during the physical demonstrations: Baseline, Planning with Detection, and Planning with Prediction. Due to the larger workspace and longer trajectories, the discrete set of time horizons at which the MPS made predictions was extended to a range of 0.1-6 s in increments of 0.1 s, with the same prediction recomputation rate as before (5 Hz).

We ran a total of 30 trials in each of the three robot modes, with a different sequence of simulated human actions occurring in each trial. Each sequence consisted of a random permutation of eight actions, with each of the four actions occurring twice. The motion of the human while performing each action in the sequence was simulated by playing back a sample trajectory of that action chosen at random from the holdout set. To further improve the realism of our simulation and model variability in human motion, we also incorporated a waiting behavior for the simulated human whenever the robot was in its path. Specifically, when the human was within the robot's safety radius and the human's approach angle towards the robot was less than 30 degrees, the simulated human would pause for a period between 4 and 5 seconds sampled from a uniform distribution, and then resume motion at 50% of the original speed until clear of the robot. This slower resumption of motion is intended to simulate a human carefully moving past the robot after the initial stop.

5.3.1 Simulation Results and Discussion

Multiple-Predictor System

We assessed the performance of the individual predictors (i.e., the VBPP, TSC, and SP methods) and the MPS with a leave-one-out cross-validation. For each iteration of the cross-validation, we held out one set of demonstrations for testing (one example of each action), 13 for training, and 6 for model selection. The overall mean prediction errors across all time horizons and iterations for the velocity-based position projection (VBPP), time series classification (TSC), sequence prediction (SP), and MPS were 181.9 cm, 43.0 cm, 174.3 cm, and 41.2 cm, respectively. We applied the Friedman test to verify that the prediction method had a significant effect on these prediction errors ($p < 0.001$, $\chi^2 = 50.22$) and then used the Wilcoxon signed rank test to perform pairwise comparisons to the MPS. Given the above averages, the MPS outperformed the VBPP and SP methods by a large margin, and exhibited a small improvement over the TSC method (VBPP and SP: $p < 0.001$; TSC: $p = 0.015$). As expected, due to the randomness of the human motion sequences, the sequence prediction method performed poorly across all time horizons. Consequently, the MPS was composed of only the other two methods, with the intuitive assignment of the VBPP method for short time horizons (0.1-0.6 s) and TSC method for the remaining time horizons (0.7-6 s).

Safety, Efficiency, and Fluency

Using the simulation, we examined various objective measures of human safety, task efficiency, and fluency of interaction. To assess the statistical significance of our results, we applied the Friedman test to determine the effect of the given robot mode (Baseline, Planning with Detection, or Planning with Prediction) on these measures, and the Wilcoxon signed rank test to assess pairwise comparisons. Results from these evaluations are summarized in Fig. 5-5 and Table 5.1.

One key measure of human safety in our evaluation is the number of times the robot's safety stop was triggered – i.e., the inability of the robotic system to anticipate

Table 5.1: Simulation Results[a]

Dependent Variable	Baseline	Planning + Detection	Planning + Prediction	Friedman Test
Safety Stop Triggers (#)	3.5 $p < 0.001$	5.1 $p < 0.001$	0.5	$\chi^2 = 50.04$ $p < 0.001$
Human Idle Time (s)	11.7 $p < 0.001$	12.6 $p < 0.001$	2.13	$\chi^2 = 28.92$ $p < 0.001$
Robot Idle Time (s)	91.7 $p < 0.001$	46.4 $p = 0.086$	51.3	$\chi^2 = 46.07$ $p < 0.001$
Human Task Time (s)	165.2 $p < 0.001$	168.1 $p < 0.001$	155.7	$\chi^2 = 27.27$ $p < 0.001$
Robot Task Time (s)	204.0 $p < 0.001$	159.6 $p = 0.002$	169.5	$\chi^2 = 51.67$ $p < 0.001$
Safety Stop Time (s)	18.1 $p < 0.001$	21.0 $p < 0.001$	2.85	$\chi^2 = 37.49$ $p < 0.001$

[a] Mean values of the dependent variables. The p values in the Baseline and Planning + Detection columns correspond with the pairwise comparisons between these modes and Planning + Prediction mode.

and avoid the movement of nearby humans. We observed that as compared to both the Baseline and Planning with Detection modes, which resulted in an average of 3.5 and 5.1 safety stop triggers, respectively, our system resulted in fewer safety stops (mean 0.5). This effect is statistically significant, and is also evident through the correlated measure of safety stop time depicted in Fig. 5-5. When compared with the Baseline mode, our system also statistically significantly shortened both idle and task times for both the human and the robot, with human and robot idle times reduced by 81.8% and 44.1%, respectively, and task times reduced by 5.8% and 16.9%, respectively. This demonstrates that not only did the incorporation of prediction and planning result in fewer safety stop triggers, but also improved task efficiency.

Compared with the Planning with Detection mode, our system also statistically significantly reduced human idle and task times by 83.1% and 7.4%, respectively. Interestingly, the Planning with Detection mode resulted in robot idle and task times comparable to our own approach. The total robot idle time is composed of the sum of time the robot stopped due to the safety system being engaged (specified as "Safety

Stop Time" in Table 5.1) and the time during which the planner commanded the robot to pause. On further inspection, we observed that the robot idle time due to safety stops is insignificant for our system as compared with that observed while the robot operated under the Planning with Detection mode. Indeed, this unplanned idle time contributed to only 5.56% of total robot idle time when using predictions, whereas its contribution increased to 45.33% when using only detections. This indicates that, although the robot remains idle using our approach for a similar amount of time as that observed in the Planning with Detection mode, this behavior was due to the planner commanding the robot to pause in order to yield to the human, improving safety and reducing human idle time.

Interleaving Prediction, Planning and Execution

We conducted an additional 30 simulation trials in which the robot planned its motion using SIPP but without interleaving planning and execution (similar to how SIPP was employed in [118]). The robot created a plan using SIPP and available predictions at the outset of its motion execution and executed it till either the goal was reached or safety stop was triggered. As compared to Planning with Prediction, this "SIPP-baseline" mode resulted in higher numbers of safety stops (5.5, $p < 0.001$) and cumulative duration (19.48 s, $p < 0.001$), higher human idle time (10.8 s, $p < 0.001$) and task time (173.0 s, $p < 0.001$), and a higher ratio of unplanned idle time (42.5%).

5.4 Evaluation of Impact of Improved Partial Trajectory Alignment on Qualities of Human-Robot Interaction in Simulation

After developing our integrated prediction and planning system and showing its benefit when compared to other methods of human-robot co-navigation, the next component of our evaluation involves analyzing the impact of improved alignment with

BEST-PTA on metrics of human-robot interaction. As I showed in Section 4.3, BEST-PTA allows for more accurate partial trajectory alignment when compared to baseline aligners (O-DTW, OSB, and MW). Furthermore, in Section 4.4 I showed that using BEST-PTA as part of the Multiple-Predictor System leads to decreased prediction error and a lower occurrence of large prediction errors. As the overall goal of this thesis is to develop robust human motion prediction methods in support of safe and efficient interaction, the next step involved evaluating how the improved prediction due to utilizing BEST-PTA for alignment translates to changes in the quality of human-robot interaction.

5.4.1 Evaluation Method

In order to carry out this analysis, I performed another set of simulations similar to that of Section 5.3. Specifically, I once again utilized the simulation of the automotive assembly plant using the AA dataset, but this time only the prediction and planning in time system was used, and the method of partial trajectory alignment was changed. Once again, 30 trials in each of the conditions were performed, with a different sequence of simulated human actions occurring in each trial. The partial trajectory aligners (BEST-PTA and the three baselines: O-DTW, OSB, and MW) were trained first with the use of the training and validation sets. Then, a separate MPS instance was trained for each of these aligners, as the alignment performance can affect the makeup of the synthesized predictor. In order to quantify the impact of changing the alignment method on the quality of human-robot interaction, I compared the same metrics of safety and efficiency as in the previous section.

5.4.2 Results and Discussion

A summary of the safety and efficiency metrics resulting from using BEST-PTA an the other alignment baselines is summarized in Table 5.2. Once again, I used a Friedman test and pairwise comparisons with the Wilcoxon signed ranks test in order to assess significance. The main effect was significant for the number of safety stop

Table 5.2: Simulation Results (BEST-PTA vs Baseline Alignment Methods)[a]

Dependent Variable	O-DTW	OSB	MW	BEST-PTA	Friedman Test
Safety Stop Triggers (#)	0.40 $p > 0.05$	0.23 $p > 0.05$	0.56 p=0.005	0.13	$\chi^2 = 8.74$ $p = 0.03$
Human Idle Time (s)	0.60	0.43	0.9	0.37	$\chi^2 = 3.35$ $p > 0.05$
Robot Idle Time (s)	34.2	33.8	38.0	35.1	$\chi^2 = 6.45$ $p > 0.05$
Human Task Time (s)	150.1 $p < 0.001$	150.4 $p < 0.001$	150.9 $p < 0.001$	147.4	$\chi^2 = 39.16$ $p < 0.001$
Robot Task Time (s)	137.3	136.1	144.0	147.8	$\chi^2 = 4.06$ $p > 0.05$
Total Task Time (s)	151.7 $p > 0.05$	152.4 $p > 0.05$	156.0 $p = 0.0014$	151.0	$\chi^2 = 19.88$ $p < 0.001$

[a] Mean values of the dependent variables. The p values in the O-DTW, OSB, and MW columns correspond with the pairwise comparisons between these modes and BEST-PTA (only shown if main effect was significant).

triggers, human task time, and total task time only. Pairwise comparisons were only significant in some of the cases (see Table 5.2 for details).

While statistical significance was not reached for all of the comparisons, the results show trends that indicate that using BEST-PTA for partial trajectory alignment has the potential for improvements in both safety and efficiency of the interaction. For example, using BEST-PTA led to fewer safety stop triggers, less human idle time, shorter human task time, and shorter overall task time.

Interestingly, BEST-PTA resulted in more robot idle time when compared to O-DTW and OSB and longer robot task time overall. One explanation for this trend is that better alignment with BEST-PTA allowed the planner to make fewer mistakes when planning adaptive behaviors. As in our simulations only the robot yields to the person, this could result in more instances of yielding behavior, resulting in more robot idle time but reduced human idle time and task time. Incorrect alignment can lead to incorrect timing of predictions being provided to the planner, which can result in the robot cutting in front of the human. If, despite the alignment being incorrect,

the robot manages to nearly avoid collision and pass in front of the human without triggering a safety stop, the robot will be able to complete its task faster than if the alignment was correct. In other instances, however, the robot will not be able to avoid collision with poor alignment, leading to the safety stop being triggered. As a result, the incorrect alignments can lead to an undesirable trading of safety for efficiency. This is also important in terms of psychological safety, as incorrect alignment causing the robot to cut in front of the person and causing many near-collisions can be very stressful to the person.

The trends in Table 5.2 indicate that BEST-PTA avoids making such a trade. Specifically, it appears that BEST-PTA, due to improved alignment, leads to better yielding behavior in that while the robot spends more time idle, the overall task time is still lower, and fewer safety stops occur. The fact that robot idle time is higher and the overall task time is lower when using BEST-PTA indicates that the risky behavior caused by incorrect alignment might allow the robot to spend less time idle, but at the expense of more safety stop triggers, which can lead to more delay than what was saved by the robot cutting in front of the human in the first place. BEST-PTA allows the planner to make the robot yield only when it is actually needed (to avoid collision), and make fewer mistakes in this regard, resulting in fewer unsafe behaviors and improved overall efficiency.

As mentioned above, the aforementioned relative results did not reach statistical significance in all instances. One possible explanation for this trend is that incorrect alignment, and therefore incorrect predictions, only have consequences in terms of safety and efficiency metrics if the incorrect alignment occurs at a few critical moments in the simulation where a motion conflict is imminent. In a majority of the task, where the human and robot are in separate sections of the workspace or moving away from each other, poor alignment would not result in any negative consequences in terms of safety and efficiency. As a result, the significance of the impact on safety and efficiency poor alignment will have depends upon the given scenario. It is likely that modifying the scenario such that motion conflicts arise more frequently would lead to

larger differences in safety and efficiency metrics than those observed in the performed simulations.

5.5 Conclusions and Future Work

In this chapter, I presented a set of evaluations aimed at assessing the feasibility and benefit of the human motion prediction approaches developed in this thesis. First, I introduced an integrated prediction and planning framework that combines the MPS with robot motion planning in time. This framework is capable of online, time-optimal planning using predictions from the MPS for a set of time horizons.

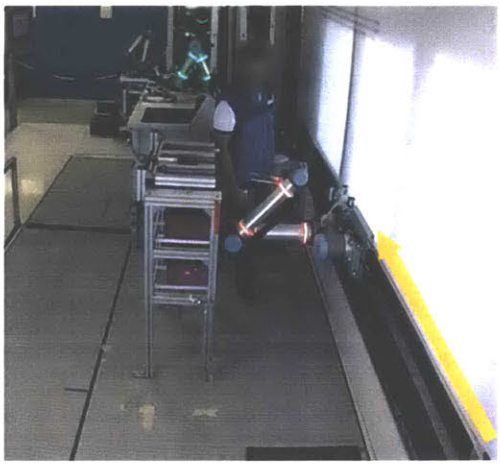
Next, I described a physical demonstration of the prediction and planning framework deployed in an emulated factory work cell. Through the physical demonstration, I was able to demonstrate the developed methods' ability to provide predictions and plan online, as well as qualitatively assess the resulting robot behavior and its impact on the interaction with the human. The demonstrations showed that by using the MPS with planning in time, the robot employs adaptive behaviors automatically, resulting in more coordinated co-navigation.

Finally, I introduced a set of simulated evaluations aimed at qualitatively assessing the impact of the developed methods' on the safety and efficiency of the interaction. Through these evaluations, I was able to show that our approach of combining predictions from the MPS with planning in time results in improvements to both safety and efficiency by, among others, lowering the number of safety stop triggers, decreasing idle time, and decreasing overall task time. By actively avoiding collisions, the integrated system improves physical safety, and by avoiding constant safety stop triggers it improves psychological safety.

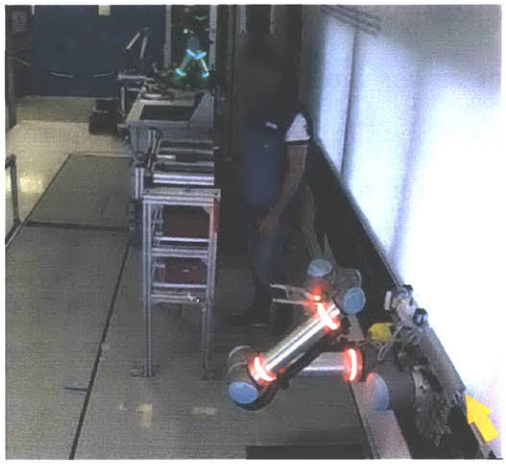
One avenue of future work is to perform integrated system evaluations of a large work cell, such as the one used in simulation, via a user study on a real physical system. An evaluation outside of simulation would provide additional fidelity and allow for an assessment of psychological safety via surveys.

(a) Start configuration.

(b) Baseline mode.



(c) Planning with detection.



(d) Planning with prediction.

Figure 5-4: Stills of the three modes (b-d) from the factory test environment demonstration. The robot's task is to navigate to the other side of the linear axis, while a human attempts to go to the depot. The black line parallel to the rail denotes the boundary of the shared region. The difference in the robot's position on the linear axis (yellow arrow) at the start (a) and between the three modes (b-d) when the human arrives at the depot illustrates the anticipatory behavior of our approach (d).

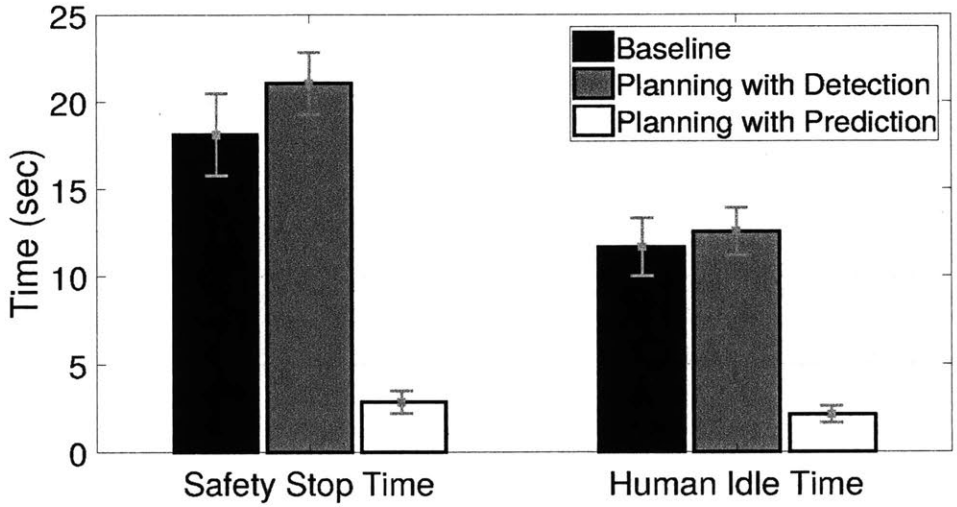


Figure 5-5: Simulation outcomes (mean and std. error) for human idle times and safety stop times (the time during which the robot was idle due to a safety stop) across the three modes.

Chapter 6

Thesis Contributions

Human motion prediction is a key enabling technology for supporting safe and efficient human-robot interaction. Due to the lack of robustness of previously developed methods, however, its deployment in real-world systems is limited. In this thesis, I introduced two novel frameworks for prediction and partial trajectory alignment in order to overcome the drawbacks of prior approaches and allow for more widespread use of human motion prediction for providing safe and efficient human-robot interaction.

First, I developed the Multiple-Predictor System (MPS), a data-driven approach for combining the complementary strengths of several component predictors. By leveraging task data directly, the MPS automatically identifies the most informative predictors and synthesizes a high-performing predictor. With the use of three distinct human motion datasets and their variants, I showed that the MPS is robust to a wide variety of task, motion, and sensing qualities.

Second, I introduced the Bayesian ESTimator for Partial Trajectory Alignment (BEST-PTA). By combining an optimization-based segmentation with temporal priors and distance-based likelihoods, I demonstrated that BEST-PTA is capable of providing accurate alignments even in the presence of trajectory overlaps and temporary stops. By evaluating the method on three human motion datasets against three baselines, I show the superior robustness of BEST-PTA. Furthermore, I demonstrated that this improvement in alignment leads to reduced prediction error when BEST-PTA is used as part of the MPS.

Lastly, I performed a variety of evaluations in order to assess the impact of the developed techniques on improving safety and efficiency of human-robot interaction. Toward this goal, I developed an integrated prediction and planning framework capable of operating online. I implemented this framework on a real physical system and demonstrated that this approach leads to adaptive robot behaviors that improve safety and efficiency. Finally, I performed a simulated evaluation through which I showed that the integrated framework leads to quantitative improvements in safety and efficiency metrics.

Bibliography

[1] Audi Smart Factory. https://www.audi-mediacenter.com/de/amtv/videos/2175. [Online].

[2] Innovative human-robot cooperation in BMW Group Production. https://www.press.bmwgroup.com/global/article/detail/T0209722EN/.

[3] Pilz Safe Camera System SafetyEye. http://www.eltron.pl/uploads/manufacturer_catalogs/16/10338/SafetyEYE_EN.pdf.

[4] Plug and Navigate robots for smart factories. http://www.pan-robots.eu/. [Online; last accessed 1-Sept-2017].

[5] Search-Based Planning Library (SBPL). http://www.sbpl.net/Software. [Online].

[6] Robonaut 2. National Aeronautics and Space Administration (NASA), https://robonaut.jsc.nasa.gov/R2/, 2019.

[7] Prasad Akella, Michael Peshkin, ED Colgate, Wit Wannasuphoprasit, Nidamaluri Nagesh, Jim Wells, Steve Holland, Tom Pearson, and Brian Peacock. Cobots for the automobile assembly line. In *ICRA*, pages 728–733. IEEE, 1999.

[8] W. Albert and T. Tullis. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics.* Interactive Technologies. Elsevier Science, 2013.

[9] T. Arai, R. Kato, and M. Fujita. Assessment of operator stress induced by robot collaboration in assembly. *CIRP Annals - Manufacturing Technology*, 59(1):5–8, January 2010.

[10] Tarik Arici, Sait Celebi, Ali S Aydin, and Talha T Temiz. Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimedia Tools and Applications*, 72(3):3045–3062, 2014.

[11] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. Measurement Instruments for the Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety of Robots. *International Journal of Social Robotics*, 1(1):71–81, November 2008.

[12] Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning Motion Patterns of People for Compliant Robot Motion. *The International Journal of Robotics Research*, 24(1):31–48, January 2005.

[13] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[14] X. Broquere, D. Sidobre, and I. Herrera-Aguilar. Soft motion trajectory planner for service manipulator robot. In *Proceedings of IROS*, pages 2808–2813. IEEE, September 2008.

[15] Maria G Bualat, Trey Smith, Ernest E Smith, Terrence Fong, and DW Wheeler. Astrobee: A new tool for iss operations. In *2018 SpaceOps Conference*, page 2517, 2018.

[16] Giovanni Buizza Avanzini, Nicola Maria Ceriani, Andrea Maria Zanchettin, Paolo Rocco, and Luca Bascetta. Safety Control of Industrial Robots Based on a Distributed Distance Sensor. 22(6):2127–2140, November 2014.

[17] John Travis Butler and Arvin Agah. Psychological Effects of Behavior Patterns of a Mobile Personal Robot. *Autonomous Robots*, 10(2):185–202, 2001.

[18] Sylvain Calinon, Irene Sardellitti, and Darwin G Caldwell. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *Proceedings of IROS*, pages 249–254. IEEE, October 2010.

[19] S. Cambon, R. Alami, and F. Gravot. A Hybrid Approach to Intricate Motion, Manipulation and Task Planning. *The International Journal of Robotics Research*, 28(1):104–126, January 2009.

[20] Don B. Chaffin and Julian J. Faraway. Stature, age and gender effects on reach motion postures. *Human Factors*, pages 408–420, 2000.

[21] Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P. How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2017.

[22] Yufan Chen, Miao Liu, Shih-Yuan Liu, Justin Miller, and Jonathan P. How. Predictive Modeling of Pedestrian Motion Patterns with Bayesian Nonparametrics. In *AIAA Guidance, Navigation, and Control Conference*, 2016.

[23] Alessandro De Luca, Alin Albu-Schäffer, Sami Haddadin, and Gerd Hirzinger. Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In *Proceedings of IROS*, pages 1623–1630, 2006.

[24] Alessandro De Luca and Fabrizio Flacco. Integrated control for phri: Collision avoidance, detection, reaction and collaboration. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 288–295. IEEE, 2012.

[25] Alessandro De Luca and Fabrizio Flacco. Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration. In *Proceedings of BioRob*, pages 288–295. IEEE, June 2012.

[26] Frédéric Dehais, Emrah Akin Sisbot, Rachid Alami, and Mickaël Causse. Physiological and subjective evaluation of a human-robot object hand-over task. *Applied ergonomics*, 42(6):785–91, November 2011.

[27] Hao Ding, G Reissig, K Wijaya, D Bortot, K Bengler, and O Stursberg. Human arm motion modeling and long-term prediction for safe and efficient Human-Robot-Interaction. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011.

[28] Simon Dixon. Live tracking of musical performances using on-line time warping. In *Proceedings of the 8th International Conference on Digital Audio Effects*, pages 92–97. Citeseer, 2005.

[29] Ana M Djuric, RJ Urbanic, and JL Rickli. A framework for collaborative robot (cobot) integration in advanced manufacturing systems. *SAE International Journal of Materials and Manufacturing*, 9(2):457–464, 2016.

[30] Christian Dondrup, Nicola Bellotto, Marc Hanheide, Kerstin Eder, and Ute Leonards. A computational model of human-robot spatial interactions based on a qualitative trajectory calculus. *Robotics*, 4(1), 2015.

[31] Shuonan Dong and Brian Williams. Learning and recognition of hybrid manipulation motions in variable environments using probabilistic flow tubes. *International Journal of Social Robotics*, 4(4):357–368, 2012.

[32] Anca D Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S Srinivasa. Effects of Robot Motion on Human-Robot Collaboration. In *Proceedings of HRI*, pages 51–58, 2015.

[33] Anca D. Dragan, Kenton C.T. Lee, and Siddhartha S. Srinivasa. Legibility and predictability of robot motion. In *Proceedings of HRI*, pages 301–308, 2013.

[34] Jos Elfring, René van de Molengraft, and Maarten Steinbuch. Learning intentions for improved human motion prediction. *Robotics and Autonomous Systems*, 62(4):591–602, April 2014.

[35] Sarah Elliott and Maya Cakmak. Robotic cleaning through dirt rearrangement planning with learned transition models. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1623–1630. IEEE, 2018.

[36] BVSA Elprama, Ilias El Makrini, and A Jacobs. Acceptance of collaborative robots by factory workers: a pilot study on the importance of social cues of anthropomorphic robots. In *International Symposium on Robot and Human Interactive Communication*, 2016.

[37] Esra Erdem, Kadir Haspalamutgil, Can Palaz, Volkan Patoglu, and Tansel Uras. Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation. In *Proceedings of ICRA*, pages 4575–4581. IEEE, May 2011.

[38] M.S. Erden and T. Tomiyama. Human-Intent Detection and Physically Interactive Control of a Robot Without Force Sensors. 26(2):370–382, April 2010.

[39] David Feil-Seifer and Maja Matarić. People-aware navigation for goal-oriented behavior involving a human partner. In *2011 IEEE International Conference on Development and Learning, ICDL 2011*, 2011.

[40] Jaime F. Fisac, Andrea Bajcsy, Sylvia L. Herbert, David Fridovich-Keil, Steven Wang, Claire J. Tomlin, and Anca D. Dragan. Probabilistically Safe Robot Planning with Confidence-Based Human Predictions. In *Robotics: Science and Systems (RSS)*, 2018.

[41] Kerstin Fischer, Lars Christian Jensen, and Leon Bodenhagen. To Beep or Not to Beep Is Not the Whole Question. In Michael Beetz, Benjamin Johnston, and Mary-Anne Williams, editors, *Proceedings of ICSR*, 2014.

[42] F. Flacco, T. Kroger, A. De Luca, and O. Khatib. A depth space approach to human-robot collision avoidance. In *Proceedings of ICRA*, pages 338–345. IEEE, May 2012.

[43] Lorenzo Fluckiger, Kathryn Browne, Brian Coltin, Jesse Fusco, Theodore Morse, and Andrew Symington. Astrobee robot software: A modern software system for space. 2018.

[44] Terrence Fong, Mark Micire, Ted Morse, Eric Park, Chris Provencher, Vinh To, D Wheeler, David Mittman, R Jay Torres, and Ernest Smith. Smart spheres: a telerobotic free-flyer for intravehicular activities in space. In *Proc. AIAA Space*, volume 13, 2013.

[45] Yoav Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, aug 1997.

[46] Milad Geravand, Fabrizio Flacco, and Alessandro De Luca. Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture. In *Proceedings of ICRA*, pages 4000–4007, 2013.

[47] Saskia Golz, Christian Osendorfer, and Sami Haddadin. Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction. In *Proceedings of ICRA*, pages 3788–3794, 2015.

[48] Birgit Graf, Matthias Hans, and Rolf D Schraft. Care-o-bot ii–development of a next generation robotic home assistant. *Autonomous robots*, 16(2):193–205, 2004.

[49] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. Soft-tissue injury in robotics. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3426–3433, May 2010.

[50] S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer. On making robots understand safety: Embedding injury knowledge into control. *IJRR*, 31(13):1578–1602, November 2012.

[51] S Haddadin, H Urbanek, S Parusel, D Burschka, J Rossmann, A Albu-Schäffer, and G Hirzinger. Real-time reactive motion generation based on variable attractor dynamics and shaped velocities. In *Proceedings of IROS*, pages 3109–3116. IEEE, October 2010.

[52] Sami Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger. Collision detection and reaction: A contribution to safe physical Human-Robot Interaction. In *Proceedings of IROS*, pages 3356–3363. IEEE, September 2008.

[53] Edward Twitchell Hall. *The Hidden Dimension.* Doubleday, 1966.

[54] Bradley Hamner, Seth Koterba, Jane Shi, Reid Simmons, and Sanjiv Singh. An autonomous mobile manipulator for assembly tasks. *Autonomous Robots*, 28(1):131–149, 2010.

[55] Kelsey P. Hawkins, Shray Bansal, Nam N. Vo, and Aaron F. Bobick. Anticipating human actions for collaboration in the presence of task and sensor uncertainty. In *Proceedings of ICRA*, pages 2215–2222. IEEE, May 2014.

[56] Kelsey P. Hawkins, Nam Vo, Shray Bansal, and Aaron F. Bobic. Probabilistic Human Action Prediction and Wait-sensitive Planning for Responsive Human-robot Collaboration, 2013.

[57] Bradley Hayes and Julie A. Shah. Interpretable models for fast activity recognition and anomaly explanation during collaborative robotics tasks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6586–6593. IEEE, may 2017.

[58] J. Heinzmann and A. Zelinsky. Quantitative Safety Guarantees for Physical Human-Robot Interaction. *IJRR*, 22:479–504, 2003.

[59] Tin Kam Ho. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE, 1995.

[60] Guy Hoffman and Cynthia Breazeal. Cost-based anticipatory action selection for human-robot fluency. In *IEEE Transactions on Robotics*, volume 23, pages 952–961, 2007.

[61] John-Paul Hosom. Speaker-independent phoneme alignment using transition-dependent states. *Speech Communication*, 51(4):352–368, 2009.

[62] Chien-Ming Huang, Maya Cakmak, and Bilge Mutlu. Adaptive Coordination Strategies for Human-Robot Handovers, 2015.

[63] Yun Jiang and Ashutosh Saxena. Modeling High-Dimensional Humans for Activity Anticipation using Gaussian Process Latent CRFs, 2014.

[64] Michiel Joosse, Aziez Sardar, Manja Lohse, and Vanessa Evers. BEHAVE-II: The Revised Set of Measures to Assess User's Attitudinal and Behavioral Responses to a Social Robot. *International Journal of Social Robotics*, 5(3):379–388, June 2013.

[65] Michiel P. Joosse, Ronald W. Poppe, Manja Lohse, and Vanessa Evers. Cultural differences in how an engagement-seeking robot should approach a group of people. In *Proceedings of the 5th ACM international conference on Collaboration across boundaries: culture, distance & technology - CABS '14*, pages 121–130, New York, New York, USA, aug 2014. ACM Press.

[66] Leslie Pack Kaelbling and Tomas Lozano-Perez. Hierarchical task and motion planning in the now. In *Proceedings of ICRA*, pages 1470–1477. IEEE, May 2011.

[67] Vasiliy Karasev, Alper Ayvaci, Bernd Heisele, and Stefano Soatto. Intent-aware long-term prediction of pedestrian motion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2543–2549. IEEE, may 2016.

[68] Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–11. SIAM, 2001.

[69] Harmish Khambhaita and Rachid Alami. A human-robot cooperative navigation planner. In *HRI*, pages 161–162. ACM, 2017.

[70] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *IJRR*, 5(1):90–98, March 1986.

[71] Yunkyung Kim and Bilge Mutlu. How social distance shapes human-robot interaction. *International Journal of Human Computer Studies*, 72(12):783–795, 2014.

[72] Rachel Kirby, Reid Simmons, and Jodi Forlizzi. COMPANION: A Constraint-Optimizing Method for Person-Acceptable Navigation. In *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pages 607–612. IEEE, sep 2009.

[73] Jyrki Kivinen and Manfred K. Warmuth. Averaging Expert Predictions. pages 153–167. Springer Berlin Heidelberg, 1999.

[74] Will Knight. Smart robots can now work right next to auto workers. *MIT Technology Review*, 17, 2013.

[75] K. L. Koay, K. Dautenhahn, S. N. Woods, and M. L. Walters. Empirical results from using a comfort level device in human-robot interaction studies. In *Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction - HRI '06*, page 194, New York, New York, USA, mar 2006. ACM Press.

[76] Soenke Kock, Jan Bredahl, Peter J Eriksson, Mats Myhr, and Kevin Behnisch. Taming the robot Better safety without higher fences. *ABB Review*, pages 11–14, April 2006.

[77] Suzan Koknar-Tezel. *Optimal Subsequence Bijection and Classification of Imbalanced Data Sets.* PhD thesis, Philadelphia, PA, USA, 2011. AAI3440089.

[78] Marina Kollmitz, Kaijen Hsiao, Johannes Gaa, and Wolfram Burgard. Time dependent planning on a layered social cost map for human-aware robot navigation. In *European Conference on Mobile Robots*, pages 1–6. IEEE, 2015.

[79] Hema S Koppula and Ashutosh Saxena. Anticipating Human Activities using Object Affordances for Reactive Robotic Response. In *Proceedings of RSS*, 2013.

[80] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743, December 2013.

[81] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. Feature-Based Prediction of Trajectories for Socially Compliant Navigation. In *Proceedings of RSS*, Sydney, Australia, 2012.

[82] Dana Kulić and Elizabeth Croft. Physiological and subjective responses to articulated robot motion. *Robotica*, 25(01):13, August 2006.

[83] Dana Kulić and Elizabeth Croft. Pre-collision safety strategies for human-robot interaction. *Autonomous Robots*, 22:149–164, 2007.

[84] Dana Kulić and Elizabeth A. Croft. Safe planning for human-robot interaction. *Journal of Robotic Systems*, 22(7):383–396, July 2005.

[85] Bakir Lacevic and Paolo Rocco. Kinetostatic danger field - A novel safety assessment for human-robot interaction. In *Proceedings of IROS*, pages 2169–2174, 2010.

[86] Matteo Laffranchi, N. G. Tsagarakis, and Darwin G. Caldwell. Safe human robot interaction via energy regulation control. In *Proceedings of IROS*, pages 35–41. IEEE, October 2009.

[87] Stephane Lallee, Katharina Hamann, Jasmin Steinwender, Felix Warneken, Uriel Martienz, Hector Barron-Gonzales, Ugo Pattacini, Ilaria Gori, Maxime Petit, Giorgio Metta, Paul Verschure, and Peter Ford Dominey. Cooperative

human robot interaction systems: IV. Communication of shared plans with Naïve humans using gaze and speech. In *Proceedings of IROS*, pages 129–136, 2013.

[88] P. A. Lasota and J. A. Shah. Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human-Robot Collaboration. *Human Factors*, 57(1):21–33, January 2015.

[89] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics*, 5(4):261–349, 2017.

[90] Przemyslaw A. Lasota, Gregory F. Rossano, and Julie A. Shah. Toward safe close-proximity human-robot interaction with standard industrial robots. In *CASE*, pages 339–344. IEEE, August 2014.

[91] Przemyslaw A Lasota and Julie A Shah. A multiple-predictor approach to human motion prediction. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2300–2307. IEEE, 2017.

[92] Longin Jan Latecki, Qiang Wang, Suzan Koknar-Tezel, and Vasileios Megalooikonomou. Optimal subsequence bijection. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 565–570. IEEE, 2007.

[93] Benjamin Letham, Cynthia Rudin, and David Madigan. Sequential event prediction. *Machine Learning*, 93(2-3):357–380, nov 2013.

[94] Kang Li and Yun Fu. Prediction of Human Activity by Discovering Temporal Sequence Patterns. 36(8):1644–1657, August 2014.

[95] Antonio M López, Juan C Alvarez, and Diego Álvarez. Walking turn prediction from upper body kinematics: A systematic review with implications for human-robot interaction. *Applied Sciences*, 9(3):361, 2019.

[96] David V Lu and William D Smart. Towards more efficient navigation for robots and humans. In *IROS*, pages 1707–1713. IEEE, 2013.

[97] Matthias Luber, Johannes A Stork, Gian Diego Tipaldi, and Kai O Arras. People tracking with human motion predictions from social forces. In *Proceedings of ICRA*, pages 464–469. IEEE, May 2010.

[98] Ruikun Luo, Rafi Hayne, and Dmitry Berenson. Unsupervised early prediction of human reaching for humanâĂŞrobot collaboration in shared workspaces. *Autonomous Robots*, pages 1–18, jul 2017.

[99] Robert Macrae and Simon Dixon. Accurate real-time windowed time warping. In *ISMIR*, pages 423–428, 2010.

[100] Robert Macrae and Simon Dixon. A guitar tablature score follower. In *2010 IEEE International Conference on Multimedia and Expo*, pages 725–726. IEEE, 2010.

[101] Jim Mainprice, E. Akin Sisbot, Leonard Jaillet, Juan Cortes, Rachid Alami, and Thierry Simeon. Planning human-aware motions using a sampling-based costmap planner. In *Proceedings of ICRA*, pages 5012–5017, 2011.

[102] Jim Mainprice and Dmitry Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *Proceedings of IROS*, pages 299–306. Ieee, November 2013.

[103] Jim Mainprice, Rafi Hayne, and Dmitry Berenson. Predicting human reaching motion in collaborative tasks using Inverse Optimal Control and iterative replanning. In *Proceedings of ICRA*, pages 885–892, 2015.

[104] Bruce S. McEwen. Stress and the Individual. *Archives of Internal Medicine*, 153(18):2093, September 1993.

[105] Ross Mead and Maja J Matarić. Autonomous human–robot proxemics: socially aware navigation based on interaction potential. *Autonomous Robots*, 41(5):1189–1201, 2017.

[106] Yoichi Morales, Atsushi Watanabe, Florent Ferreri, Jani Even, Tetsushi Ikeda, Kazuhiro Shinozawa, Takahiro Miyashita, and Norihiro Hagita. Including human factors for planning comfortable paths. In *Proceedings of ICRA*, pages 6153–6159, 2015.

[107] Rainer Müller, Matthias Vette, and Matthias Scholer. Robot workmate: A trustworthy coworker for the continuous automotive assembly line and its implementation. *Procedia CIRP*, 44:263–268, 2016.

[108] Jonathan Mumm and Bilge Mutlu. Human-robot proxemics: Physical and Psychological Distancing in Human-Robot Interaction. In *Proceedings of HRI*, page 331, 2011.

[109] Stefanos Nikolaidis, Keren Gu, Ramya Ramakrishnan, Julie Shah, and R O May. Efficient Model Learning for Human-Robot Collaborative Tasks. pages 1–9, 2015.

[110] Stefanos Nikolaidis, Przemyslaw Lasota, Ramya Ramakrishnan, and Julie Shah. Improved human–robot team performance through cross-training, an approach inspired by human team training practices. *The International Journal of Robotics Research*, 34(14):1711–1730, 2015.

[111] Stefanos Nikolaidis, Przemyslaw Lasota, Gregory Rossano, Carlos Martinez, Thomas Fuhlbrigge, and Julie Shah. Human-robot collaboration in manufacturing: Quantitative evaluation of predictable, convergent joint action. In *IEEE ISR 2013*, pages 1–6. IEEE, October 2013.

[112] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007.

[113] Tatsuya Nomura, Tomohiro Suzuki, Takayuki Kanda, and Kensuke Kato. Measurement of negative attitudes toward robots. *Interaction Studies*, 7(3):437–454, 2006.

[114] Kazuhide Okamoto, Karl Berntorp, and Stefano Di Cairano. Similarity-based vehicle-motion prediction. In *2017 American Control Conference (ACC)*, pages 303–308. IEEE, 2017.

[115] Jae Sung Park, Chonhyon Park, and Dinesh Manocha. Intention-Aware Motion Planning Using Learning Based Human Motion Prediction. *Robotics science and systems (RSS)*, 2017.

[116] Jae Sung Park, Chonhyon Park, and Dinesh Manocha. Intention-aware motion planning using learning based human motion prediction. In *R:SS*, 2017.

[117] Claudia Pérez-D'Arpino and Julie A Shah. Fast Target Prediction of Human Reaching Motion for Cooperative Human-Robot Manipulation Tasks using Time Series Classification. In *Proceedings of ICRA*, 2015.

[118] Mike Phillips and Maxim Likhachev. SIPP: Safe interval path planning for dynamic environments. In *ICRA*. IEEE, 2011.

[119] Erion Plaku and Gregory D Hager. Sampling-Based Motion and Symbolic Action Planning with geometric and differential constraints. In *Proceedings of ICRA*, pages 5002–5008. IEEE, May 2010.

[120] Matteo Parigi Polverini, Andrea Maria Zanchettin, and Paolo Rocco. Real-time collision avoidance in human-robot interaction based on kinetostatic safety field. In *Proceedings of IROS*, pages 4136–4141. IEEE, September 2014.

[121] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[122] "robust". Merriam-webster.com, 2019.

[123] Francesco Rovida and Volker Krüger. Design and development of a software architecture for autonomous mobile manipulators in industrial environments. In *International Conference on Industrial Technology*, pages 3288–3295. IEEE, 2015.

[124] Andrey Rudenko, Luigi Palmieri, Achim J Lilienthal, and Kai O Arras. Human motion prediction under social grouping constraints. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3358–3364. IEEE, 2018.

[125] Paul Rybski, Peter Anderson-Sprecher, Daniel Huber, Chris Niessl, and Reid Simmons. Sensor fusion for human safety in industrial workcells. In *Proceedings of IROS*, pages 3612–3619. IEEE, October 2012.

[126] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1036–1043, 2011.

[127] M S Ryoo, Thomas J Fuchs, Lu Xia, J K Aggarwal, and Larry Matthies. Robot-Centric Activity Prediction from First-Person Videos : What Will They Do to Me ? In *Proceedings of HRI*, pages 295–302, 2015.

[128] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, Feb 1978.

[129] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[130] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[131] Rolf Dieter Schraft, Christian Meyer, Christopher Parlitz, and Evert Helms. Powermate-a safe and intuitive robot assistant for handling and assembly tasks. In *ICRA*, pages 4074–4079. IEEE, 2005.

[132] Sara Sheikholeslami, AJung Moon, and Elizabeth A Croft. Cooperative gestures for industry: Exploring the efficacy of robot hand configurations in expression of instructional gestures for human–robot interaction. *IJRR*, 2017.

[133] E. A. Sisbot and R. Alami. A Human-Aware Manipulation Planner. 28(5):1045–1057, October 2012.

[134] E.A. Sisbot, L.F. Marin-Urias, R. Alami, and T. Simeon. A Human Aware Mobile Robot Motion Planner. 23(5):874–883, October 2007.

[135] Emrah Akin Sisbot, Luis F. Marin-Urias, Xavier Broquère, Daniel Sidobre, and Rachid Alami. Synthesizing Robot Motions Adapted to Human Presence. *International Journal of Social Robotics*, 2(3):329–343, June 2010.

[136] Aaron St. Clair and Maja Mataric. How Robot Verbal Feedback Can Improve Team Performance in Human-Robot Task Collaborations. In *Proceedings of HRI*, pages 213–220, New York, New York, USA, March 2015. ACM Press.

[137] Nicholas Stergiou. *Innovative analyses of human movement.* Human Kinetics, The address, 1 edition, 2004.

[138] Daniel Szafir, Bilge Mutlu, and Terrence Fong. Communication of Intent in Assistive Free Flyers. In *Proceedings of HRI*, pages 358–365, 2014.

[139] Daniel Szafir, Bilge Mutlu, and Terrence Fong. Communicating Directionality in Flying Robots. In *Proceedings of HRI*, pages 19–26, 2015.

[140] Wataru Takano, Hirotaka Imagawa, and Yoshihiko Nakamura. Prediction of human behaviors in the future through symbolic inference. In *Proceedings of ICRA*, pages 1970–1975, 2011.

[141] Leila Takayama, Doug Dooley, and Wendy Ju. Expressing thought: Improving Robot Readability with Animation Principles. In *Proceedings of HRI*, page 69, 2011.

[142] Leila Takayama and Caroline Pantofaru. Influences on proxemic behaviors in human-robot interaction. In *Proceedings of IROS*, pages 5495–5502, 2009.

[143] Wenjun Tan, Chengdong Wu, Shuying Zhao, and Jiang Li. Dynamic hand gesture recognition using motion trajectories and key frames. *2010 2nd International Conference on Advanced Computer Control*, 3:163–167, 2010.

[144] Paolo Tormene, Toni Giorgino, Silvana Quaglini, and Mario Stefanelli. Matching incomplete time series with dynamic time warping: An algorithm and an application to post-stroke rehabilitation. *Artificial intelligence in medicine*, 45:11–34, 12 2008.

[145] Vaibhav V Unhelkar, P Claudia, Leia Stirling, and Julie A Shah. Human-Robot Co-Navigation using Anticipatory Indicators of Human Walking Motion. In *Proceedings of ICRA*, 2015.

[146] Vaibhav V Unhelkar, Stefan Dorr, Alexander Bubeck, Przemyslaw A Lasota, Jorge Perez, Ho Chit Siu, James C Boerkoel, Quirin Tyroller, Johannes Bix, Stefan Bartscher, et al. Mobile robots for moving-floor assembly lines: Design, evaluation, and deployment. *IEEE Robotics & Automation Magazine*, 25(2):72–81, 2018.

[147] Vaibhav V Unhelkar, Przemyslaw A Lasota, Quirin Tyroller, Rares-Darius Buhai, Laurie Marceau, Barbara Deml, and Julie A Shah. Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. *IEEE Robotics and Automation Letters*, 3(3):2394–2401, 2018.

[148] Vaibhav V Unhelkar, JM Perez, James C Boerkoel, Johannes Bix, Stefan Bartscher, and Julie A Shah. Towards control and sensing for an autonomous mobile robotic assistant navigating assembly lines. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4161–4167. IEEE, 2014.

[149] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[150] A. Vick, D. Surdilovic, and J. Krüger. Safe physical human-robot interaction with industrial dual-arm robots. In *9th International Workshop on Robot Motion and Control, RoMoCo 2013 - Workshop Proceedings*, pages 264–269, 2013.

[151] Christian Vogel, Christoph Walter, and Norbert Elkmann. A projection-based sensor system for safe physical human-robot collaboration. In *Proceedings of IROS*, pages 5359–5364, 2013.

[152] Jacob Walker, Abhinav Gupta, and Martial Hebert. Patch to the Future: Unsupervised Visual Prediction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3302–3309. IEEE, June 2014.

[153] Eduard Wall, Lars Schillingmann, and Franz Kummert. Online nod detection in human-robot interaction. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 811–817. IEEE, 2017.

[154] Michael L. Walters, Mohammedreza A. Oskoei, Dag Sverre Syrdal, and Kerstin Dautenhahn. A long-term Human-Robot Proxemic study. In *2011 RO-MAN*, pages 137–142. IEEE, jul 2011.

[155] Jason Wolfe, B. Marthi, and Stuart Russell. Combined task and motion planning for mobile manipulation. *ICAPS*, 2010.

[156] Michał Woźniak, Manuel Graña, and Emilio Corchado. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17, 2014.

[157] Huanmei Wu, Betty Salzberg, Gregory C Sharp, Steve B Jiang, Hiroki Shirato, and David Kaeli. Subsequence matching on structured time series data. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 682–693. ACM, 2005.

[158] Shuang Xiao, Zhan Wang, and John Folkesson. Unsupervised robot learning to predict person motion. In *Proceedings of ICRA*, pages 691–696, 2015.

[159] Aras Yurtman and Billur Barshan. Detection and evaluation of physical therapy exercises by dynamic time warping using wearable motion sensor units. In *Information Sciences and Systems 2013*, pages 305–314. Springer, 2013.

[160] Andrea Zanchettin, Andrea Casalino, Luigi Piroddi, and Paolo Rocco. Prediction of human activity patterns for human-robot collaborative assembly tasks. *IEEE Transactions on Industrial Informatics*, 2018.

[161] Andrea Maria Zanchettin, Nicola Maria Ceriani, Paolo Rocco, Hao Ding, and Bjorn Matthias. Safety in Human-Robot Collaborative Manufacturing Environments: Metrics and Control. pages 1–12, 2015.

[162] Feng Zhou and Fernando De la Torre. Generalized time warping for multi-modal alignment of human motion. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1282–1289. IEEE, 2012.

[163] Brian D. Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J. Andrew Bagnell, Martial Hebert, Anind K. Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *Proceedings of IROS*, pages 3931–3936, 2009.