Human Intention Inference using Fusion of Gaze and Motion Information

Daniel Trombetta
daniel.trombetta@uconn.edu

Follow this and additional works at: https://opencommons.uconn.edu/gs_theses

Recommended Citation
https://opencommons.uconn.edu/gs_theses/1549

This work is brought to you for free and open access by the University of Connecticut Graduate School at OpenCommons@UConn. It has been accepted for inclusion in Master's Theses by an authorized administrator of OpenCommons@UConn. For more information, please contact opencommons@uconn.edu.
Human Intention Inference using Fusion of Gaze and Motion Information

Daniel Trombetta

B.S., University of Connecticut, 2018

A Thesis
Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science at the University of Connecticut

2020
Human Intention Inference using Fusion of Gaze and Motion Information

Presented by
Daniel Trombetta, B.S.

Major Advisor
Ashwin P. Dani

Associate Advisor
Yaakov Bar-Shalom

Associate Advisor
Liang Zhang

University of Connecticut
2020
I would like to extend my gratitude to the many people without whom this achievement would not have been possible. First and foremost, I’d like to sincerely thank my major advisor, Dr. Ashwin P. Dani for his willingness to offer his wisdom and guidance to me. His support and patience have fostered a healthy learning environment and motivated me through the tough times. It has been a pleasure to have the opportunity to work with you. I’d like to thank the members of my committee, Dr. Yaakov Bar-Shalom and Dr. Liang Zhang, for their insightful suggestions and valuable input towards my work.

I’m grateful for the friends I’ve made in the Robotics and Controls group. We’ve been through some great times and certainly some difficult ones. I never could have succeeded without your constant support throughout the many courses, projects, papers, experiments, and long nights.

I am forever indebted to my parents and grandparents. I never would have made it here without your undying love and support. To my brother, sister, and friends back home, I thank you for always keeping my spirits up and believing in me unconditionally.
Contents

1 Introduction 1
   1.0.1 Motivation .................................................. 1
   1.0.2 Related Work ............................................... 3
1.1 Background ......................................................... 4
   1.1.1 A Review of Coordinate Systems and Transformations .... 5
   1.1.2 Some Common Motion Models ................................ 8
   1.1.3 Kalman Filtering and Extended Kalman Filtering ....... 10
1.2 Outline of the Dissertation ....................................... 12
1.3 Publications ........................................................ 14

2 Fusion of Pupil and Hand Motion for Human Intention Inference 16
   2.1 Introduction ....................................................... 16
   2.2 Problem Formulation and Solution Approach .................. 17
   2.3 Eye Gaze Data Processing ....................................... 18
   2.4 Motion Models ..................................................... 20
      2.4.1 Human Hand Motion ........................................ 20
      2.4.2 Human Eye-Gaze Motion ................................... 22
   2.5 Estimation of Human Intention .................................. 23
      2.5.1 Eye-gaze Filter .............................................. 24
      2.5.2 Hand Motion Filter With Gaze Fusion .................... 24
   2.6 Experimental Results ............................................. 28

3 Human Intention Estimator with Variable Structure 33
   3.1 Introduction ......................................................... 33
   3.2 Problem Formulation ............................................... 34
   3.3 Motion Models ..................................................... 35
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.1 Human Hand Motion</td>
<td>35</td>
</tr>
<tr>
<td>3.3.2 Eye-gaze Motion</td>
<td>35</td>
</tr>
<tr>
<td>3.4 Model Set Augmentation</td>
<td>36</td>
</tr>
<tr>
<td>3.4.1 Definition of Model Sets</td>
<td>36</td>
</tr>
<tr>
<td>3.4.2 Model Set Adaptation Using Reachable Workspace Constraints</td>
<td>38</td>
</tr>
<tr>
<td>3.4.3 Model Set Adaptation Using Human Visual Span</td>
<td>40</td>
</tr>
<tr>
<td>3.4.4 Addition and Removal of Active Models</td>
<td>40</td>
</tr>
<tr>
<td>3.5 VS-IMM Human Intention Inference Algorithm</td>
<td>41</td>
</tr>
<tr>
<td>3.5.1 Human Hand Motion Filter</td>
<td>42</td>
</tr>
<tr>
<td>3.5.2 Eye-gaze Filter</td>
<td>44</td>
</tr>
<tr>
<td>3.5.3 Determination of Human Intention</td>
<td>45</td>
</tr>
<tr>
<td>3.6 Discussion</td>
<td>47</td>
</tr>
<tr>
<td>3.7 Experiment</td>
<td>48</td>
</tr>
<tr>
<td>4 Conclusion and Future Work</td>
<td>53</td>
</tr>
<tr>
<td>Appendix A Additional Derivations</td>
<td>55</td>
</tr>
<tr>
<td>A.1 Learning Contracting Human Dynamics</td>
<td>55</td>
</tr>
<tr>
<td>Bibliography</td>
<td>58</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 A single cycle of the Kalman filter ........................................ 11
1.2 A single cycle of the extended Kalman filter ........................ 13
2.1 Overview of the Hyperface CNN architecture ...................... 19
2.2 A block diagram to summarize the data acquisition, processing, intention estimation, and hand motion prediction for the proposed algorithm. 28
2.3 Five frames from the RGB video collected by the Kinect sensor each overlaid with a bounding box around the current predicted goal object. 30
2.4 The top plot compares the evolution of all six model probabilities where vertical dotted lines denote when the current most likely model has changed. The lower plot compares the true model with the predicted model. ..................................................... 31
2.5 Hand position tracked by the IMM filter using fused model probabilities. 32
3.1 The simplified human arm model is shown above. Joints \( J_1 \) and \( J_2 \) are not used in this work. The point \( S \) denotes the shoulder position as detected by the Kinect sensor with the positive directions of the Kinect coordinate system attached. ................. 39
3.2 This diagram summarizes the HIEVS algorithm ...................... 46
3.3 Two RGB images taken from the Kinect sensor corresponding to frames 10 and 105. The images have been overlain with bounding boxes around the face and a vector from the face to the predicted gaze point. 49
3.4 The human gaze point tracked by the HIEVS algorithm is shown in dotted red. The measurements, in solid blue, are acquired by passing the RGB images collect by the Kinect sensor through a deep network for gaze point prediction. ........................................ 50
3.5 Human hand motion tracked by the HIEVS algorithm is shown in dotted red. The hand motion data is acquired via the Kinect sensor's skeletal tracking feature. ........................................ 51
3.6 The evolution of the fused model probabilities associated with the $N$ models. The vertical dotted black line denotes the change in true intention. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
Enabling robots with the ability to quickly and accurately determine the intention of their human counterparts is a very important problem in Human-Robot Collaboration (HRC). The focus of this work is to provide a framework wherein multiple modalities of information, available to the robot through different sensors, are fused to estimate a human’s action intent. In this thesis, two human intention estimation schemes are presented. In both cases, human intention is defined as a motion profile associated with a single goal location. The first scheme presents the first human intention estimator to fuse information from pupil tracking data as well as skeletal tracking data during each iteration of an Interacting Multiple Model (IMM) filter in order to predict the goal location of a reaching motion. In the second, two variable structure IMM (VS-IMM) filters, which track gaze and skeletal motion, respectively, are run in parallel and their associated model probabilities fused. This method is advantageous over the first as it can be easily scaled to include more models and provides greater disparity between the most likely model and the other models. For each VS-IMM filter, a model selection algorithm is proposed which chooses the most likely models in each iteration based on physical constraints of the human body. Experimental results are provided to validate the proposed human intention estimation schemes.
Chapter 1

Introduction

1.0.1 Motivation

In recent years, Human Robot Collaboration (HRC) has gained an increasing amount of interest, especially in regards to its applications to manufacturing, surgical, and rehabilitation fields. HRC, as its name suggests, refers to scenarios in which a human works directly with a robot without the presence of safety barriers or cages. The collaboration may involve physical interaction between the human and the robot, which is often referred to in the literature as Human Robot Interaction (HRI), or it may not not involve any physical interactions.

HRC is sought after in assembly line scenarios as it has the potential to decrease the costs associated with purchasing and maintaining many machines dedicated to performing a single task. It can also allow for a human-robot team to perform tasks which would be much too dexterous for a machine to perform. In the surgical field, the use of human-robot teams shows promise in increasing success rates as robots do
not tire in the manner in which humans do and can perform tasks with great levels of repeatability. The use for robotics for rehabilitation would allow for care to be tailored to each user’s unique capabilities.

In order for such human-robot teams to be successful, however, a major step must first be achieved: the determination of the human’s intention. When collaborating with a human, it is vital for both the safety of the human in the absence of safety barriers and for the completion of the objective that the robot is capable of inferring the human’s current action intention. Once the action intention is determined, the robot can select the appropriate counter-action that it should take.

In this work, the focus is on the inference of human intention. Clearly, human intention is not a directly measurable phenomena and, thus, must be estimated through the use of measurable quantities. In the literature, several definitions of human intention exist and are typically driven by the data available or the objective to be performed. Several of these will be discussed in the next section. In this thesis, human intention is defined as a motion profile which converges to a single goal location. This definition is motivated by the fact that in the three sample scenarios given previously, tasks which are completed by the human, i.e. lifting components, making an incision, etc., are often characterized completely by the properties of the motion involved and a single location of interest. While most of the work in the literature focuses on determining human intention through the use of a single modality of information at a time, this thesis describes how multiple modalities of data, namely gaze and motion data, can be fused in order to provide a robot with an accurate early estimate of human intention.
1.0.2 Related Work

The goal of HRC is to have humans and robots work seamlessly with one another. A clear model to follow then is that of collaboration between humans. Studies have shown that when two humans interact, they infer one another’s intent in order to safely and effectively collaborate.\cite{1,25}. This finding was extended to collaboration between human and robots in \cite{15,15,27} where it was shown that inference of the human’s intention improves the overall performance of the task. Various different modalities of information about the human have been used in order to infer a human’s intention.

In \cite{9}, characteristics of the objects in the workspace are used to predict which of a finite list of high-level activities, such as stacking objects, picking objects, etc., a human is performing. Human movement is used in \cite{17} to predict trajectories in manipulation tasks. Various types of physiological information has been used for intention prediction. Electromyography is used for human intention classification in \cite{22}. Heart rate and skin response are used in \cite{10} to predict a humans affective state and common a robot partner accordingly. Intention inference as a goal-reaching motion profile estimation for collaboratively carrying heavy objects is presented in \cite{21}. In \cite{13}, a Gaussian Process (GP) is used to predict hand trajectories during an object handover task.

In recent years, the measurement and estimation of 3-dimensional (3D) human eye gaze has gained attention specifically for use in human intention inference. In \cite{30}, it is shown that a human’s gaze is directly related to their intended actions. It is demonstrated that adults predict action goals by fixating on the end location of an action even before it is reached in \cite{5}. In \cite{7}, it is shown that gaze communicates
attention. Consequently, several intention inference schemes have adopted use of gaze information for human intention estimation. In [24], a robot assistant is provided a control input based on a human’s intention measure acquired from the human’s gaze. The intention to handover an object is predicted by using key features extracted from the vision and the pose (position + orientation) data in [26]. In [20], a convolution neural network (CNN) model is used to predict the human gaze from an RGB image in order to initialize model probabilities for an Interacting Multiple Model (IMM) filter [2] and track human hand motion. This method, however, does not utilize gaze information after initialization.

While many of the human intention inference algorithms which currently exist in the literature leverage only a single modality of data, the algorithms proposed in this thesis utilize two modalities of information, namely human gaze and motion data, which are measured and fused in every iteration in order to produce an accurate early prediction of human intention.

1.1 Background

In this section, some preliminary material which is used throughout this thesis is presented. First, a review of coordinate systems is given and the process of representing data with respect to various coordinate frames is described. Then, in order to provide a basis for common estimation methods, a few common motion models used in basic estimation schemes are described in a general form. Two general estimation schemes, the Kalman Filter (KF) and a derivative of the KF for nonlinear dynamics known as the Extended Kalman Filter (EKF), are presented in the following subsection.
1.1.1 A Review of Coordinate Systems and Transformations

Coordinate systems are a mathematical tool used in geometry to assign vectors to points and objects in a given space to define their relative positions and orientations, or pose. When the pose of an object is measured, it is said to be measured with respect to a certain coordinate system, or reference frame. Reference frames can be placed however the user sees the most fit. Moreover, in a given scene, there could exist many reference frames. An common example of this observed in this work is that of reference frames attached to various sensors. When pose measurements are given by a calibrated sensor, they are given with respect to that sensor’s internal reference frame. In general, when working in 3-dimensional (3D) space, at least 6 values are needed to describe an object’s pose: 3 for the position and 3 for orientation. There are several ways to represent the pose of an object, however, for the purpose of this thesis, this section describes only the homogeneous transformation matrix.

In order to define the homogenous transformation matrix, it is important to first introduce the orthonormal rotation matrix, or just rotation matrix, representation of orientation. Adopting the notation from [4], a rotation matrix which transforms the description of a vector defined with respect to a frame B to a vector with respect to frame A is given as

\[
\begin{bmatrix}
A_x \\
A_y \\
A_z
\end{bmatrix} = ^A R_B \begin{bmatrix}
B_x \\
B_y \\
B_z
\end{bmatrix}
\]  

(1.1)

where \( ^A R_B \in \mathbb{R}^{3 \times 3} \) is the rotation matrix that maps the vector in frame B to frame A and the superscripts \( A \) and \( B \) denote the frame with which the vector is defined.
with respect to. A 3D rotation matrix $^A R_B$ has the following properties:

1. it is orthonormal as each of its columns is a unit vector and are orthogonal with one another

2. the columns are the unit vectors that define the axes of the rotated frame $B$ with respect to $X$ and are by definition both unit-length and orthogonal.

3. it belongs to the special orthogonal group of dimension 3 or $R \in SO(3) \subset \mathbb{R}^{3 \times 3}$. This means that the product of any two matrices within the group also belongs to the group, as does its inverse.

4. its determinant is +1, which means that the length of a vector is unchanged after transformation, that is, $||^Y p|| = ||^X p||, \forall \theta$

5. the inverse is the same as the transpose, that is, $R^{-1} = R^T$

   Rotations of an angle $\theta$ about the $x-$, $y-$, and $z-$ axes are given by

   $$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (1.2)$$

   $$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (1.3)$$

   $$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$
Rotations about various axes can then be combined simply by multiplying rotation matrices with one another. For example, by convention, when describing the attitude of aircraft, roll-pitch-yaw angles or Tait-Bryan angles are used which generally use the sequence \( ZYX \). The rotation matrix for a roll-pitch-yaw angle representation of rotation is then given by

\[
R = R_z(\theta_{yaw})R_y(\theta_{pitch})R_x(\theta_{roll}) \tag{1.5}
\]

Using this representation of orientation, the homogeneous transformation matrix can be defined. Recall that a homogeneous transformation matrix describes the pose, both position and orientation, of one frame with respect to another frame. The form of a homogeneous transformation matrix is given as

\[
\begin{bmatrix}
A_x \\
A_y \\
A_z \\
1
\end{bmatrix} = \begin{bmatrix}
A_R \\
t
0_{1 \times 3}
\end{bmatrix}
\begin{bmatrix}
B_x \\
B_y \\
B_z \\
1
\end{bmatrix} = A\tilde{p} = \begin{bmatrix}
A_R \\
t
0_{1 \times 3}
\end{bmatrix}B\tilde{p} \tag{1.6}
\]

where \( A_R \) defines the orientation of frame \( B \) with respect to \( A \), \( t \) is the vector defining the origin of \( B \) with respect to \( A \), and \( A\tilde{p} \) and \( B\tilde{p} \) are said to be the homogeneous form of the point \( p \) with relative to frames \( A \) and \( B \), respectively.

Like the rotation matrix, homogeneous transformation matrices can be multiplied together to define linear transformations through multiple reference frames as

\[
A\tilde{p} = A_T B_T C_T \tilde{p} \tag{1.7}
\]
where \( B^T_C \) and \( A^T_B \) are the homogeneous transformation matrices which transform a point in frame \( C \) to frame \( B \) and from \( B \) to \( A \), respectively. This mathematical tool will be used in this thesis to represent the acquired data with respect to various different reference frames.

### 1.1.2 Some Common Motion Models

In order to utilize the filtering techniques described in Section 1.1.3, models are needed to describe the expected motion of the target. The two models used in this work are augmentations of two common motion models seen in the literature known as the discrete white noise acceleration model and the discrete wiener process acceleration model.

The discrete white noise acceleration (DWNA) model is used to model motion which has a constant velocity between sampling points and a piecewise constant white noise acceleration. The state equation for this second order model is given by

\[
x(k + 1) = Fx(k) + \Gamma \nu(k)
\]

where \( \nu(k) \) is a zero-mean white acceleration sequence, the transition matrix is given as

\[
F = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}
\]

where \( T_s \) is the sample time, and the vector gain which multiplies the scalar process
noise is given as

\[
\Gamma = \begin{bmatrix}
\frac{1}{2}T_s^2 \\
T_s
\end{bmatrix}
\]  

(1.10)

in light of the fact that the increment in velocity and position between samples are \(T_s \nu(k)\) and \(\frac{1}{2}T_s^2 \nu(k)\).

The covariance of \(\Gamma \nu(k)\) is given as

\[
Q = \begin{bmatrix}
\frac{1}{4}T_s^4 & \frac{1}{2}T_s^3 \\
\frac{1}{2}T_s^3 & T_s^2
\end{bmatrix} \sigma_\nu^2
\]  

(1.11)

where \(\sigma_\nu^2\) is the covariance of the process \(\nu(k)\), and both \(\nu(k)\) and \(\sigma_\nu^2\) have units of \(\frac{\text{length}}{\text{time}^2}\).

The discrete Wiener process acceleration (DWPA) model also has a white process noise \(\nu(k)\), but in this case it refers to the acceleration increment. The state equation is

\[
x(k + 1) = F x(k) + \Gamma \nu(k)
\]  

(1.12)

where here

\[
F = \begin{bmatrix}
1 & T_s & \frac{1}{2}T_s^2 \\
0 & 1 & T \\
0 & 0 & 1
\end{bmatrix}
\]  

(1.13)

\[
R = \begin{bmatrix}
\frac{1}{2}T_s^2 \\
T \\
1
\end{bmatrix}
\]  

(1.14)
and the covariance of the process noise multiplied by the gain is

\[
Q = \begin{bmatrix}
\frac{1}{4}T_s^4 & \frac{1}{2}T_s^3 & \frac{1}{2}T_s^2 \\
\frac{1}{2}T_s^3 & T_s^2 & T \\
\frac{1}{2}T_s^2 & T & 1
\end{bmatrix}
\] (1.15)

1.1.3 Kalman Filtering and Extended Kalman Filtering

The Kalman filter is a recursive algorithm used for dynamic estimation. Under the assumption that the initial state and all of the noises in the system are Gaussian distributed, the Kalman filter is the optimal minimum mean squared error estimator. If the random variables are not Gaussian, but one still has access to their first two moments, then the Kalman filter is the best linear state estimator. Using the Markov assumption, it is sufficient to know only the state estimate and its associated covariance at time \( k \) rather than the entire history prior to time \( k \). One cycle of the Kalman filter consists of mapping

\[
\hat{x}(k|k) \triangleq E[x(k)|Z^k] P(k|k) = E[[x(k) - \hat{x}(k|k)][x(k) - \hat{x}(k|k)]']|Z^k]
\] (1.16)

where \( \hat{x}(k|k) \) and \( P(k|k) \) are the state estimate and covariance at time \( k \) given the information available up to time \( k \), and \( Z^k \) are the set of \( k \) measurements, to the corresponding variables in the following stage, \( \hat{x}(k+1|k+1) \) and \( P(k+1|k+1) \). The Kalman filter makes the assumptions that the initial state estimate and covariance is known and all noises in the system are uncorrelated with each other and the initial state. The computation of a single cycle of the Kalman filter is shown in Figure 1.1.

For non-linear systems, a variation of the Kalman filter, called the extended
Figure 1.1: A single cycle of the Kalman filter
Kalman filter (EKF) is used. The EKF is obtained through a Taylor series expansion of the nonlinear dynamics and measurement equations. For a system with dynamics

\[ x(k + 1) = f[k, x(k), u(k)] + \nu(k) \]  

(1.17)

and measurement equation

\[ z(k) = h[k, x(k)] + w(k) \]  

(1.18)

where \( f[k, x(k), u(k)] \) and \( h[k, x(k)] \) are the nonlinear dynamics and measurements, respectively, a single cycle of the first order EKF, obtained by taking the first order Taylor series expansion around the latest estimate \( \hat{x}(k|k) \), is given in Figure 1.2

1.2 Outline of the Dissertation

This thesis is organized as follows: In Chapter 2 an algorithm is presented which uses the fusion of hand and gaze motion data to estimate a human’s action intention. The class of problems to which this intention estimation method should be applied is described and the solution approach is described in a general manner. Then, a method for transforming data, which is taken with respect to a dynamic reference frame with uncertain motion, into a static reference frame is described. The motion models which are used to describe the hand and gaze motion are presented. Finally, the human intention estimation algorithm is presented and validated with a real world experiment. In Chapter 3 the Human Intention Estimator with Variable Structure (HIEVS) algorithm is presented. Again, the problem is formulated and motion models
Figure 1.2: A single cycle of the extended Kalman filter
introduced. Then, two model set augmentation algorithms are presented which are used to select the active models sets for two interacting multiple model filters with variable structures. The HIEVS algorithm is presented. The chapter is concluded with an experimental evaluation of the algorithm and a discussion of the results. A summary of the contents of this thesis and the algorithms presented is given in Chapter 4.

1.3 Publications

Conference papers that are accepted and published with primary authorship include:


Conference papers that are accepted and published with co-authorship include:


Magazine articles that are accepted and published with co-authorship include:


Journal papers with primary authorship in preparation:

Chapter 2

Fusion of Pupil and Hand Motion for Human Intention Inference

2.1 Introduction

This chapter introduces an algorithm which fuses data acquired through pupil tracking and skeletal tracking in order to determine a human’s current action intention. Human intention is defined here as a motion profile which converges to a single goal location. As mentioned in Section 1.0.2 many of the existing human intention estimation schemes utilize only a single modality of data. Moreover, while the algorithm presented in [20] does leverage gaze data in addition to motion data, the gaze data is merely used to initialize prior probabilities for a hand tracking IMM and the remainder of the algorithm relies solely on motion data for the intention inference. In contrast, the algorithm presented in this chapter leverages both pupil tracking data, which is used to estimate the human’s 3D gaze point, and motion data which is
acquired using skeletal tracking in order to determine the human’s intention.

2.2 Problem Formulation and Solution Approach

Many HRC scenarios are geared towards tasks that require that a series of subtasks are first completed in order to successfully complete the overarching task. For example, in an assembly task, the individual components must be gathered and pieced together in order to complete an assembly. Often times, the subtasks are not necessarily required to be completed in any specific order. In these cases, it is vital that each member involved in the collaboration is made aware of the current step that the other member is performing. Consider a situation wherein a human and a robot are working collaboratively to complete a task of this description. Each task is associated with a model which consists of a motion profile that terminates at exactly one goal location. The goal locations are defined as the positions of the task-relevant objects in the workspace. In an initial training phase, several expert demonstrations are given of each step in the task in order to train a single layer neural network (NN) under contraction constraints and effectively teach the robot the motion profiles associated with each model. The only additional information known to the robot at the onset of the objective is the goal location of each model. The human partner progresses through the sequence in any order which they see fit. The robot does not have knowledge of the sequence a priori but must be able to infer which of the learned tasks is currently being completed. During the operation, the robot partner collects measurements of the human partner’s current hand motion and gaze point location. Using this information, the robot must infer which model the human is operating
under, or moreover, which task the human is currently performing. A Microsoft Kinect Sensor is used to track the 3-dimentional (3D) position of the human’s skeleton in the Kinect frame, $F_K$, and Pupil Glasses by Pupil Labs \[6\] worn by the human are used to acquire 3D estimates of the human’s gaze locations in the Pupil glasses frame, $F_P$. The transformation of the gaze points into $F_K$ can be obtained using the 3D location of the human’s head as tracked by the Kinect sensor in conjunction with the Hyperface CNN, which can detect faces in RGB images and predict the roll, pitch, and yaw orientations.

### 2.3 Eye Gaze Data Processing

For each measured 3D gaze point $A_P \in \mathbb{R}^3$ in the Pupil glasses frame $F_P$, it is required to transform $A_P$ into frame $F_K$ as, firstly, the positions of the goal locations are defined in $F_K$, and more importantly, it is desirable to have the gaze point measurements with respect to a static frame. $F_P$, however, is dynamic with respect to the goal locations because the glasses are being worn by a mobile human. In order to represent $A_P$ with respect to frame $F_K$, the homogeneous transformation matrix, introduced in Section \[1.1.1\] which maps reference frame $F_P$ to $F_K$ must be determined. Let the origin of $F_P$ be approximated by the 3D coordinates of the human head, $X_{head} \in \mathbb{R}^3$, measured by the Kinect sensor in $F_K$. Then the translation component of the homogeneous transformation matrix can be represented by $X_{head}$. The rotation component can be obtained using the Hyperface CNN. Hyperface is a CNN architecture that takes an RGB image of any size as its input and returns whether or not the image contains one or more faces, places landmarks on relevant points of
Figure 2.1: Overview of the Hyperface CNN architecture.

the faces, provides a measure of the visibility of the landmarks, gives estimates of the roll, pitch, and yaw of each face in the image in its own reference frame $F_H$, and predicts the gender associated with each face. Hyperface is trained on annotated images provided by the Annotated Facial Landmarks in the Wild (AFLW) dataset [8]. The AFLW dataset contains about 25,000 faces within real-world images. Each face is annotated with 21 facial landmarks, the gender, the roll-pitch-yaw angles, and a visibility measure. The general structure of the Hyperface CNN architecture is shown in Figure 2.1. The RBG images of size $640 \times 480$ collected by the Kinect sensor are used as input for the Hyperface CNN in order to obtain the roll, pitch, and yaw estimates of the human’s head in $F_H$. In order to utilize this information, the point $A_P$ must first be transformed into the Hyperface frame $F_H$ using $A_H = H R_P A_P$, where
where $P_H$ denotes a point in the the Hyperface frame $F_H$ and $^H R_P \in \mathbb{R}^{3 \times 3}$ denotes the rotation matrix from frame $F_H$ to $F_P$. The complete transformation can then be given as

$$A_K = {}^K R_H ^H R_P A_P + X_{head},$$

$$T^K_P = \begin{bmatrix} {}^K R_H ^H R_P & X_{head} \\ 0_{1 \times 3} & 1 \end{bmatrix}$$

where $^K R_H \in \mathbb{R}^{3 \times 3}$ denotes the rotation matrix from $F_K$ to $F_H$ and is obtained using the prediction of the human head orientation from Hyperface, and $T^K_P \in \mathbb{R}^{4 \times 4}$ is the homogeneous transformation that maps the point $[(A_P)^T, 1]^T$ to the point $[(A_K)^T, 1]^T$.

### 2.4 Motion Models

In this section, human hand motion and human eye-gaze motion models are described in detail.

#### 2.4.1 Human Hand Motion

At any given time, the human is assumed to be operating according to one of $N$ models. Let $G = [g_1, g_2, \ldots, g_N]$ represent the vector of all $N$ goal locations. Then, the $i^{th}$ model $M_i$ is associated with a single goal location $g_i$. Each model is characterized by the motion of the human hand as well as the evolution of the gaze point. The human hand motion associated with the $i^{th}$ model is a modified version of the DWPA
model described in Section 1.1.2 given by

\[
\begin{bmatrix}
  x_{H}(k+1) \\
  \dot{x}_{H}(k+1) \\
  \ddot{x}_{H}(k+1)
\end{bmatrix} = \\
\begin{bmatrix}
  \text{diag}_3(1) & \text{diag}_3(T_s) & \text{diag}_3(\frac{1}{2}T^2_s) \\
  0 & \text{diag}_3(1) & \text{diag}_3(T_s) \\
  0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
  x_H(k) \\
  \dot{x}_H(k) \\
  \ddot{x}_H(k)
\end{bmatrix} \\
+ \\
\begin{bmatrix}
  0 \\
  0 \\
  f_i(x_H(k), \dot{x}_H(k), \ddot{x}_H(k))
\end{bmatrix} \\
+ \\
\begin{bmatrix}
  W_1 \\
  W_2 \\
  W_3
\end{bmatrix}
w_1(k)
\]  

(2.3)

where \( x_H \in \mathbb{R}^3 \) is the 3-dimentional (3D) position of the human hand, \( T_s \) is the sampling time, the operator \( \text{diag}_\eta(\rho) \) denotes a square matrix of dimension \( \eta \times \eta \) with the value \( \rho \) along the central diagonal, \( f_i : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3 \) is a continuously differentiable function modelling the \( i^{th} \) model’s acceleration, \( W_1 = \text{diag}_3(\frac{1}{6}T^3_s) \), \( W_2 = \text{diag}_3(\frac{1}{2}T^2_s) \), \( W_3 = \text{diag}_3(T_s) \), and \( w_1 \sim \mathcal{N}(0, Q_1) \) is a Gaussian distributed process noise with zero mean and known covariance \( Q_1 \in \mathbb{R}^{3 \times 3} \) that represents the model uncertainty in acceleration update. Each function \( f_i \) is approximated by a neural network whose parameters are learned from data collected during the training phase. The training is performed subject to a contraction metric which garunteees, even with minimal training data, that predictions made by each \( f_i \) are stable and exponentially converge to the \( i^{th} \) goal location. For a more detailed look at the training method, the reader is referred to Section A.1. The noisy measurements of the human partner’s
hand positions are modeled as

$$z_H(k) = x_H(k) + \nu_1(k)$$ (2.4)

where $\nu_1(k) \in \mathbb{R}^{3 \times 3}$ is a Gaussian distributed measurement noise with zero mean and known covariance $R_1 \in \mathbb{R}^{3 \times 3}$.

### 2.4.2 Human Eye-Gaze Motion

Unlike the hand motion, which has a distinct motion model for each goal location, there is a single model for the gaze as the behavior of the eye motion is expected to be similar regardless of goal location. The evolution of the human’s gaze point is modeled as a constant velocity model, introduced in Section 1.1.2 and is given by

$$
\begin{bmatrix}
  x_E(k + 1) \\
  \dot{x}_E(k + 1)
\end{bmatrix}
= 
\begin{bmatrix}
  \text{diag}_3(1) & \text{diag}_3(T_s) \\
  0 & \text{diag}_3(1)
\end{bmatrix}
\begin{bmatrix}
  x_E(k) \\
  \dot{x}_E(k)
\end{bmatrix}
+ 
\begin{bmatrix}
  W_4 \\
  W_5
\end{bmatrix} w_2(k)
$$ (2.5)

where $x_E \in \mathbb{R}^3$ is the 3D position of the human’s gaze, $W_4 = \text{diag}_3(\frac{1}{2}T_s^2), W_5 = \text{diag}_3(T_s)$, and $w_2 \sim \mathcal{N}(0, Q_2)$ is a Gaussian distributed process noise with zero mean and known covariance $Q_2 \in \mathbb{R}^{3 \times 3}$ that represents model uncertainties in velocity update. The measurement model is given as

$$z_E(k) = x_E(k) + \nu_2(k)$$ (2.6)
where \( \nu_2(k) \in \mathbb{R}^{3 \times 3} \) is a Gaussian distributed measurement noise with zero mean and known covariance \( R_2 \in \mathbb{R}^{3 \times 3} \).

### 2.5 Estimation of Human Intention

Consider the \( N \) models, \( M_1, M_2, \ldots, M_N \), with goal locations \( g_1, g_2, \ldots, g_N \) and the measurement models defined in (3.2) and (2.6). Let
\[
X_H = [x_H^T, \dot{x}_H^T, \ddot{x}_H^T]^T, \quad X_E = [x_E^T, \dot{x}_E^T]^T
\]
denote the human hand and eye-gaze state vectors, respectively, and
\[
Z_{H}^{1:k} = [z_H(1), z_H(2), \ldots, z_H(k)], \quad Z_{E}^{1:k} = [z_E(1), z_E(2), \ldots, z_E(k)]
\]
denote a set of \( k \) measurements of the human hand and the eye-gaze, respectively. The algorithm introduced in this section fuses the information obtained from the measurements of the gaze point and the human hand in order to infer which of the \( N \) models the human is currently operating under and effectively compute the state estimate \( \hat{X}_H(k|k) \). Note that the true model that the human is operating under is not known to the robot and the human could switch among the \( N \) models at any time. The formulation is separated into two subsections. First, a Kalman filter (KF) that estimates the current gaze point is presented. The gaze point estimates are used to calculate probabilities that the human is operating according to each model. The second is an IMM filter for human hand motion which uses \( N \) extended Kalman filters (EKFs) running in parallel to filter the hand motion. At the beginning of each iteration of the IMM filter, the model posterior probabilities produced in the previous iteration are fused with those from the eye-gaze filter to generate more informative model probabilities.
2.5.1 Eye-gaze Filter

Using the dynamics in (2.5) and the noisy measurements in (2.6), a Kalman filter is designed to obtain estimates of the gaze point and the corresponding covariances. Once two measurements are available, the filter is initialized using the two-point differencing method. Each iteration, \( \hat{x}_E \) and \( S \), the eye-gaze filter’s state estimate and innovation covariance, respectively, are obtained. The probability that the current estimated gaze point is associated with the model having a goal location \( g_j \) can be represented as

\[
\mu_j^E(k) = P(\hat{x}_E(k) | g_j) = \frac{1}{\sqrt{2\pi S}} e^{-\frac{1}{2}(\hat{x}_E - g_j)^T S^{-1} (\hat{x}_E - g_j)}
\]  

(2.7)

2.5.2 Hand Motion Filter With Gaze Fusion

Once the first model probability \( \mu_j^E(0) \) is made available by the eye-gaze filter, the prior probabilities for the IMM \( \mu_j^H(0) \) can be initialized as \( \mu_j^H(0) = \mu_j^E(0) \). In subsequent iterations of the filter, the model probabilities from each filter \( \mu_j^H(k) \) and \( \mu_j^E(k) \) are fused according to

\[
\mu_j^F(k) = \alpha e^{-\beta T_t} \mu_j^E(k) + (1 - \alpha e^{-\beta T_t}) \mu_j^H(k)
\]  

(2.8)

where \( T_t \in [0, \infty) \) denotes the time since the most recent model switch, \( \alpha \in [0, 1] \) is a user defined parameter that determines the degree to which the weight shifts between the model probabilities from the two filters over time, and \( \beta \in (0, \infty) \) is a user defined parameter which controls how quickly the weight shift occurs. \( \alpha \) should be chosen by the user proportional their belief of how informative the gaze information is at
the beginning of a subtask. For example, if the task requires the user to perform a complicated visual search, causing their gaze to fluctuate rapidly before settling on their goal, then \( \alpha \) should be set high. \( \beta \) should be chosen by the user to reflect the speed of the tasks. For example, if pieces of an assembly are very close to one another, it is likely that the human hand can move among them quickly, and \( \beta \) should be set to a higher value to allow the algorithm to account for the rapid motion and rapidly shift the balance of the weights. Dynamically weighing the model probabilities can account for situations wherein one source is expected to provide more reliable insight. In general, one would want \( \mu^F_j(k) \) to hold a higher weight at the beginning of a task because it evolves quickly relative to hand motion. However, toward the end of the objective, \( \mu^H_j(k) \) should hold more weight because, ultimately, the algorithm is predicting the goal locations of reaching tasks, and it is possible that the gaze point has shifted although the current goal location has not yet changed.

**Interaction/Mixing:** At the beginning of each iteration, the initial conditions (state estimate \( \hat{x}_H^{0j}(k-1|k-1) \) and covariance \( \hat{P}_H^{0j}(k-1|k-1) \)), where superscript 0 denotes initial condition, \( j \) denotes the number of the filter, at time \( k \), are adjusted by mixing the filter outputs from the previous iteration (time instant \( k-1 \)) in the following way

\[
\hat{x}_H^{0j}(k-1|k-1) = \sum_{i=1}^{N} \hat{x}_H^{ij}(k-1|k-1) \\
	imes \mu_{i,j}^F(k-1|k-1), j = 1, ..., N
\] (2.9)
\[ \hat{P}^{0j}_H(k-1|k-1) = \sum_{i=1}^{N} \mu^F_{ij}(k-1|k-1) \hat{P}^i_H(k-1|k-1) \]

\[ + \left[ \hat{x}^i_H(k-1|k-1) - \hat{x}^{0j}_H(k-1|k-1) \right] \times \left[ \hat{x}^i_H(k-1|k-1) - \hat{x}^{0j}_H(k-1|k-1) \right]^T \]

\[ j = 1, \ldots, N \] (2.10)

where \( \hat{x}^i_H(k-1|k-1) \) and \( \hat{P}^i_H(k-1|k-1) \) are the state estimate and its covariance respectively corresponding to model \( M_j \) at time \( k-1 \) and the mixing probabilities \( \mu^F_{ij}(k-1|k-1) \) are given by

\[ \mu^F_{ij}(k-1|k-1) = \frac{\Pi_{ij}\mu^F_i(k-1)}{\bar{c}_j}, \ i, j = 1, 2, \ldots, N \] (2.11)

where \( \Pi_{ij} = p(M(k) = M_j|M(k-1) = M_i) \) is the model transition or jump probability and \( \mu^F_i(k-1) = p(M_i|Z^{1:k-1}_H, Z^{1:k-1}_E) \) is the fused probability of \( i^{th} \) model \( M_i \) being the right model at time \( k-1 \) and \( \bar{c}_j = \sum_{i=1}^{N} \Pi_{ij}\mu^F_i(k-1) \) are the normalizing constants.

**Model Matched Filtering:** Once the initial conditions \( \hat{x}^{0j}_H(k-1|k-1) \) and \( \hat{P}^{0j}_H(k-1|k-1) \) are available for each filter, the state estimate and its covariance for each model are computed using the EKFs matched to the models. Along with the state estimates and the corresponding covariances, the likelihood functions \( \Lambda_j(k) \) are computed using the mixed initial condition (3.7) and the corresponding covariance (3.8). The likelihood \( \Lambda_j(k) \), a Gaussian distribution with the predicted measurement
as the mean and the covariance equal to the innovation covariance, is given by

$$
\Lambda_j(k) = p(z_H(k)|M_j(k), Z_H^{1:k-1})
$$

$$
\Lambda_j(k) = \mathcal{N}(z_H(k); \hat{z}_H^j(k|k-1); \hat{\bar{x}}_H^0j(k-1|k-1)),
$$

$$
S_H^j(k; \hat{P}_H^0j(k-1|k-1))), \ j = 1, \ldots, N
$$

(2.12)

where $S_H^j(k; \hat{P}_H^0j(k-1|k-1))$ is the innovation covariance and $\hat{z}_H^j(k|k-1; \hat{\bar{x}}_H^0j(k-1|k-1))$ is the $j$th filter’s predicted measurement at time $t$.

**Model Probability Update:** After the likelihood functions of the models $\Lambda_j(k)$ are available, the model posterior probabilities $\mu_H^j(k)$ are calculated as follows

$$
\mu_H^j(k) = P(g_j|Z_H^{1:k}) = P(M_j(k)|Z_H^{1:k})
$$

$$
\mu_H^j(k) = p(z_H(k)|M_j(k), Z_H^{1:k-1})P(M_j(k)|Z_H^{1:k-1})
$$

$$
\mu_H^j(k) = \frac{\Lambda_j(k)\bar{c}_j}{\sum_{i=1}^N \Lambda_i(k)\bar{c}_i}, \quad j = 1, 2, \ldots, N
$$

(2.13)

and the goal location estimate $\hat{g}(t)$ is given by

$$
\hat{g}(k) = \arg\max_{g \in G} \mu_H^j(k)
$$

(2.14)

The optimization problem in (2.14) is solved by choosing the location $g_i \in G$ corresponding to the model $M_i$ with the highest model probability $\mu_H^j(k)$ at time $k$. Figure 2.2 summarizes the gaze and motion fusion algorithm in the form of a block diagram.
2.6 Experimental Results

In order to validate the utility of the proposed data fusion method, an experiment is designed in which a human partner must complete a set of tasks in any order. Using available measurements of the human hand motion and eye gaze location, the robot must determine the sequence of tasks being completed on the fly. This experimental structure is analogous to real life collaborative tasks such as two workers collaboratively hammering nails into a board at multiple locations, carrying a heavy object from one location to one of many possible destinations, or manufacturing a product that may have leniency in the order in which it is assembled, i.e. an electrical circuit. For this experiment, six goal objects, \( N = 6 \), are used: a hammer, a screwdriver, pliers, two wood blocks of different sizes, and a cardboard box. The objects are placed arbitrarily within the field of view of a Kinect sensor at known locations in \( F_K \). The human first grabs one of the tools at random, and relocates it atop any one of the three boxes as they see fit. The human’s 3D gaze point is
determined using noisy measurements provided by Pupil glasses worn by the human partner and the filter described in (2.3). Noisy measurements of the human hand motion are made available by the Kinect motion sensor’s skeletal tracking feature according to (3.2). The gaze point can be represented in the Kinect frame using (2.1). The objective is to show that the proposed algorithm can predict which tool the human is reaching for before it is grasped, and determine where it will be placed before it reaches that point. The results of this experiment are shown in the following section. Figure 2.3 shows frames from the RGB video acquired by the Kinect at relevant time instances, namely, the first and last frames along with each frame in which a model switch was predicted. A bounding box has been overlaid on each image around the object which the algorithm believes is the goal location at the current time instance. The parameter $\alpha$ from (2.8) was chosen to be 0.6 and $\beta$ was chosen to be 1 meaning that whenever a model change is predicted, the fused model probabilities are weighted 60% on $\mu^E$ and only 40% on $\mu^H$. The more time spent operating under the same model, the more weight that is shifted to $\mu^H$. As a direct result of this, the prediction of the goal location at the onset of the experiment is correct even though the human hand is not yet near the target object. As the subject reaches for the first goal location, i.e., the screwdriver, their gaze begins to move towards the next goal location. This causes the model prediction to change slightly before the true model changes and once again gives $\mu^E$ a higher weight. The incorrect intermediate predictions seen in frames 22 and 46 are due to the gaze point traveling over the objects between the previous goal and the current goal. Frame 76 shows that the correct model prediction is made 35 frames before the screwdriver is actually placed on the wood board in frame 111. Figure 2.4 shows the performance of the algorithm over time. The top plot shows the evolution of each of the six model probabilities
associated with each object in the experiment. The vertical dotted lines denote the
times when the algorithm predicts that the current model has changed. The bottom
graph shows a comparison between the true intention and the estimated intention.
The hand position tracked by the IMM filter using the fused model probabilities can
be seen in Figure 2.3.

![Five frames from the RGB video collected by the Kinect sensor each overlaid with a bounding box around the current predicted goal object.](image)

**Figure 2.3:** Five frames from the RGB video collected by the Kinect sensor each overlaid with a bounding box around the current predicted goal object.
Figure 2.4: The top plot compares the evolution of all six model probabilities where vertical dotted lines denote when the current most likely model has changed. The lower plot compares the true model with the predicted model.
Figure 2.5: Hand position tracked by the IMM filter using fused model probabilities.
Chapter 3

Human Intention Estimator with Variable Structure

3.1 Introduction

The algorithm presented in this chapter, called Human Intention Estimator with Variable Structure (HIEVS), is an extension of the work presented in Chapter 2. The algorithm from Chapter 2 uses a fixed structure IMM (FS-IMM) filter in conjunction with a Kalman filter in order to produce model probabilities from two different sources of data. Due to the FS-IMM filter, which runs all $N$ filters matched to the $N$ total models on every iteration, the computational complexity increases quickly with $N$. Thus, $N$ is restricted to few models. To overcome this scaling issue, the work in this paper utilizes variable structure IMM (VS-IMM) filter. The VS-IMM algorithm, presented in [29, 23, 28], is similar in structure to a FS-IMM except that at the beginning of each iteration, a model set augmentation (MSA) algorithm selects the most
likely subset of the total model set. The filters are only run for models corre-
spending to the active model set. By constraining the active models in the model set, a
large number of total possible intention models can be considered without necessarily
increasing the computational burden.

3.2 Problem Formulation

Consider a scenario wherein a human and a robot are collaboratively performing
an assembly task in a large warehouse environment. At the onset of the operation,
the human and the robot are both aware of all \(N\) components in the assembly, the
location of the components within the warehouse, and how to attach the component
to the assembly. It is assumed that each component is defined by its 3D coordinates
within the warehouse, defined as \(G = [g_1, g_2, ..., g_N]\) and is associated with exactly
one building instruction, or motion profile, that is taught to the robot via expert
demonstrations. As in Chapter 2, a model is considered as a motion profile and
goal location pair where \(M = [M_1, M_2, ..., M_N]\) is the entire set of \(N\) models. The
assembly process is such that components can be attached in many different sequences
to achieve the desired result. The human begins to assemble the components in a
sequence which they see fit. The robot is equipped with 3-dimensional (3D) skeletal
tracking data of the human and 3D gaze point measurements. With this information,
the robot must determine the current step the human is performing so that it may
take the appropriate action.
3.3 Motion Models

This section presents the human hand motion and human eye-gaze motion models used as well as the measurement models.

3.3.1 Human Hand Motion

At any given time, the human is assumed to be operating according to one of \(N\) models in \(M\). The human hand motion associated with the \(i^{th}\) model is given by Equation 2.3. The measurement model is given by Equation 3.2.

3.3.2 Eye-gaze Motion

The evolution of the human’s gaze-point associated with the \(i^{th}\) model is given by the discretized point attractor dynamics

\[
\begin{bmatrix}
  x_E(k+1) \\
  \dot{x}_E(k+1)
\end{bmatrix} =
\begin{bmatrix}
  1 & T_s \\
  -K_P & 1 - K_D T_s
\end{bmatrix}
\begin{bmatrix}
  x_E(k) \\
  \dot{x}_E(k)
\end{bmatrix}
+ \begin{bmatrix}
  0 \\
  K_{1g}T_s
\end{bmatrix} + \begin{bmatrix}
  W_2 \\
  W_3
\end{bmatrix} w_2(k)
\]

(3.1)

where \(x_E \in \mathbb{R}^3\) is the (3D) gaze-point, \(K_P\) and \(K_D\) are scalar gains learned in a training phase, and \(w_2 \sim \mathcal{N}(0, Q_2)\) is a Gaussian distributed process noise with zero mean and known covariance \(Q_2 \in \mathbb{R}^{3 \times 3}\) that represents the model uncertainty in acceleration update. The noisy measurements of the human partner’s eye-gaze
positions are modeled as

$$z_E(k) = x_E(k) + \nu_2(k)$$

(3.2)

where $\nu_2(k) \in \mathbb{R}^{3 \times 3}$ is a Gaussian distributed measurement noise with zero mean and known covariance $R_2 \in \mathbb{R}^{3 \times 3}$.

### 3.4 Model Set Augmentation

When designing VS-IMM filters, proper selection of a method to augment the current model set being considered by the VS-IMM is vital for the success of the algorithm. In the remainder of this section, two novel MSA algorithms are described which select subsets of the entire model set $M$ based on two physical properties of humans: reachable workspace and visual span.

#### 3.4.1 Definition of Model Sets

Model sets are defined as follows:

- $M = [M_1, M_2, ..., M_N]$ is the complete model set of $N$ possible models. Each model $M_i$ is associated with a single motion profile and goal location $g_i$ where $G = [g_1, g_2, ..., g_N]$. Due to the definition of human intention in this work and the one-to-one relationship of models to goal locations, the terms model, intention, and goal location can be used interchangeably in this paper.

- $M^E_a(k)$ is the active model set available to the eye-gaze filter at time $k$

- $M^E_i(k)$ is the inactive model set reserved by the eye-gaze filter at time $k$
• $M_a^H(k)$ is the active model set available to the hand motion filter at time $k$

• $M_r^H(k)$ is the inactive model set reserved by the hand motion filter at time $k$

At any given instance,

$$M_a^E(k) \cap M_r^E(k) = M_a^H(k) \cap M_r^H(k) = \emptyset$$

$$M_a^E(k) \cup M_r^E(k) = M_a^H(k) \cup M_r^H(k) = M$$

In order to utilize the VS-IMM framework, a valid MSA technique must be chosen. In [12], it is stated that in general, an MSA approach should possess the following properties.

1. It provides a general criterion for model activation and termination. The criterion serves as a general measure of the closeness between the true mode and the candidate models with different structures or parameters.

2. It is computationally feasible. The MSA process can be applied easily with an acceptable computational burden. This property is especially important for models characterized by continuous parameters. It requires that the MSA algorithm should provide a scheme to generate new models from the continuous mode space.

3. It is independent of filters. This requirement allows the MSA algorithm to depend only on the models themselves, and thus can exclude effects of various filters.

The two MSA algorithms proposed in the remainder of this section to select the active model sets, $M_a^H$ and $M_a^E$, are designed with the properties above in mind.
3.4.2 Model Set Adaptation Using Reachable Workspace Constraints

The MSA algorithm designed to select the active models available to the hand motion filter during each iteration $M^H_a$, leverages the simple fact that objects which lie outside of the region that is immediately reachable by the human given their position within the workspace are not likely to be the current goal. Thus, models whose goal locations lie outside of the reachable workspace need not be considered in the filter. A discussion on the implications of this assumption is provided in Section 3.6.

A method presented in [14, 11] is utilized to evaluate the human’s reachable workspace. A six degree-of-freedom (DoF) model of the human arm, comprised of six revolute joints, is used in conjunction with the associated joint limits of a healthy subject to determine all possible points which are reachable by the human’s wrist relative to their static shoulder. By evaluating this region before the onset of the intention inference process and assuming it to be constant relative to the shoulder joint, which is also tracked during the process by the skeletal tracker, the goal locations which lie within the reachable region can be determined in each iteration without bearing much computational load.

In order to estimate the reachable workspace, denoted $R_{ws}$, the six DoF human arm model is simplified further to a four DoF model by eliminating the two degrees of freedom which model flexion-extension and abduction-adduction in the inner shoulder joint and produce minimal effects in wrist position. The simplified model is shown in Figure 3.1 with the eliminated joints shaded in grey. Each joint is sampled at 25 points between their joint limits in order to obtain a map of possible wrist positions. The region $R_{ws}$ is then defined as the volume bounded by the outer-most wrist positions.
Figure 3.1: The simplified human arm model is shown above. Joints J₁ and J₂ are not used in this work. The point S denotes the shoulder position as detected by the Kinect sensor with the positive directions of the Kinect coordinate system attached.

In the map. Due to the anatomical properties of the arm, bones and muscles, some joint limits are dependent on the positions of the other joint angles. The joint limits are given as

\[
q_1 \in [-9^\circ, 160^\circ] \\
q_2 \in [(-43 + \frac{q_1}{3})^\circ, (153 - \frac{q_1}{6})^\circ] \\
q_3 \in [(-90 + \frac{7q_1}{9} - \frac{q_2}{9} + \frac{2q_1q_2}{810})^\circ, \\
(90 + \frac{4q_1}{9} - \frac{5q_1}{9} + \frac{5q_1q_2}{810})^\circ]
\]

In each iteration, the active model set available to the hand-tracking filter is then chosen to be

\[
M^H_a = \mathcal{R}_{ws} \cap G = \mathcal{R}_{ws} \cap M
\]
3.4.3 Model Set Adaptation Using Human Visual Span

In the vision literature, the human visual span, or peripheral span, refers to the region of the visual field from which one can extract information during an eye fixation. Similar to the logic used above, if a goal location does not lie within the visual span, it is unlikely to be the true goal location. Again, the implications of this assumption will be discussed in Section 3.6. In [18], a series of experiments are performed to measure the human visual span during an object search in real-world scenes. The results show that the average visual span during an object search is a cone whose aperture has a radius of 8 degrees.

Let $\vec{vs}$ be the vector from the position of the human’s eyes, estimated as $X_{head}$, to the current gaze point with respect to the pupil glasses reference frame $A_P$. Then, for the eye-gaze filter, the models that are chosen to be active at time $k$ are those which fall within the region $R_{vs}$ defined with respect to $F_K$ as the volume of a cone centered about $\vec{vs}$ with a radius of 8 degrees.

That is

$$M_{a}^{E} = R_{vs} \cap G = R_{vs} \cap M \quad (3.5)$$

3.4.4 Addition and Removal of Active Models

When a model which was active in the previous iteration becomes inactive, its corresponding model probability is set to be zero, and the EKF corresponding to the model is made inactive. That is, when the MSA algorithm determines that a model $M_i$ should be moved from set $M_a^{H}$ to set $M_r^{H}$ at time $k$, then $\mu_i^{H} = 0$, and the $i^{th}$ filter does not run on the $k^{th}$ iteration. The same logic holds true for the eye-gaze
On the other hand, when a model which was previously inactive to both filters and thus had a model probability $\mu^F = 0$ at time $k - 1$ becomes active at time $k$, it’s fused model probability is initialized to a small threshold value $\pi_{th}$ and its associated filter is made active for the $k^{th}$ iteration. If $\omega$ models which were previously inactive become active at time $k$, then their associated model probabilities are initialized equally as

$$\mu^F_\omega = \frac{\pi_{th}}{\omega} \quad (3.6)$$

where $\mu^F_\omega$ are the model probabilities associated with the set of $\omega$ models which were activated at time $k$. It is important to reiterate that the initialization described above needs to be performed if and only if $M_i \in M^H_i$ and $M_i \in M^E_i$. Otherwise, the associated fused model probability will already be non-zero.

### 3.5 VS-IMM Human Intention Inference Algorithm

In this section, the filtering step of the Human Intention Estimator with Variable Structure (HIEVS) algorithm is presented. The algorithm consists of two VS-IMM filters running in parallel. One filter processes eye-gaze data in order to produce an estimate of the eye-gaze point $\hat{x}_E(k)$ and the set of posterior model probabilities conditioned on gaze point measurements associated with each model $\mu^E_i(k)$. The other filter processes hand motion data in order to produce an estimate of the hand position $\hat{x}_H(k)$ and the set of posterior model probabilities conditioned on the hand position measurements associated with each model $\mu^H_i(k)$. The initial state and covariance for each filter, i.e. $\hat{x}_H(0|0), P_H(0|0)$ and $\hat{x}_E(0|0), P_E(0|0)$, are acquired using the
two-point differencing method. On the first iteration for each filter, the prior model probabilities \( \mu^F_j(0) \) are initialized to be uniform across all models in \( M \). Subsequent model probabilities \( \mu^F_j(k) \) are acquired from the fusion equation \[2.8\] defined at the end of this section.

### 3.5.1 Human Hand Motion Filter

The hand position VS-IMM filter is described below.

**Interaction/Mixing:** At the beginning of each iteration, the initial conditions (state estimate \( \hat{x}_{H}^{0j}(k-1|k-1) \) and covariance \( \hat{P}_{H}^{0j}(k-1|k-1) \)), where superscript 0 denotes initial condition, \( j \) denotes the number of the filter, at time \( k \), are adjusted by mixing the filter outputs from the previous iteration (time instant \( k-1 \)) in the following way

\[
\begin{align*}
\hat{x}_{H}^{0j}(k-1|k-1) &= \sum_{i=1}^{N} \hat{x}_{H}^{i}(k-1|k-1) \\
&\quad \times \mu^F_{ij}(k-1|k-1), j = 1, \ldots, N \\
\hat{P}_{H}^{0j}(k-1|k-1) &= \sum_{i=1}^{N} \mu^F_{ij}(k-1|k-1) \hat{P}_{H}^{i}(k-1|k-1) + \left( \hat{x}_{H}^{i}(k-1|k-1) - \hat{x}_{H}^{0j}(k-1|k-1) \right)^{T} \\
&\quad \times \left( \hat{x}_{H}^{i}(k-1|k-1) - \hat{x}_{H}^{0j}(k-1|k-1) \right) \\
&\quad j = 1, \ldots, N
\end{align*}
\] (3.7)

where \( \hat{x}_{H}^{i}(k-1|k-1) \), \( \hat{P}_{H}^{i}(k-1|k-1) \) are the state estimate and its covariance.
respectively corresponding to model $M_j$ at time $k-1$ and the mixing probabilities $\mu_{i|j}^F(k-1|k-1)$ are given by

$$\mu_{i|j}^F(k-1|k-1) = \frac{\Pi_{ij} \mu_i^F(k-1)}{c_j}, \quad i, j = 1, 2, ..., N \quad (3.9)$$

where $\Pi_{ij} = p(M(k) = M_j|M(k-1) = M_i)$ is the model transition or jump probability and $\mu_i^F(k-1) = p(M_i|Z^{1:k-1}_H, Z^{1:k-1}_E)$ is the fused probability of $i^{th}$ model $M_i$ being the right model at time $k-1$ and $c_j = \sum_{i=1}^N \Pi_{ij} \mu_i^F(k-1)$ are the normalizing constants.

**Model Matched Filtering:** Once the initial conditions $\hat{x}_0^{Hj}(k-1|k-1)$ and $\hat{P}_0^{Hj}(k-1|k-1)$ are available for each filter, the state estimate and its covariance for each model are computed using the EKFs matched to the models. Along with the state estimates and the corresponding covariances, the likelihood functions $\Lambda_j^H(k)$ are computed using the mixed initial condition \([3.7]\) and the corresponding covariance \([3.8]\). The likelihood $\Lambda_j^H(k)$, a Gaussian distribution with the predicted measurement as the mean and the covariance equal to the innovation covariance, is given by

$$\Lambda_j^H(k) = \mathcal{N}(z_H(k); \hat{z}_H^j(k|k-1; \hat{x}_0^{Hj}(k-1|k-1)), S_H^j(k; \hat{P}_0^{Hj}(k-1|k-1))) \quad j = 1, ..., N \quad (3.10)$$

where $S_H^j(k; \hat{P}_0^{Hj}(k-1|k-1))$ is the innovation covariance and $\hat{z}_H^j(k|k-1; \hat{x}_0^{Hj}(k-1|k-1))$ is the $j^{th}$ filter’s predicted measurement at time $t$.

**Model Probability Update:** After the likelihood functions of the models $\Lambda_j^H(k)$
are available, the model posterior probabilities $\mu_j^H(k)$ are calculated as follows

$$
\mu_j^H(k) = P(g_j|Z_{H}^{1:k}) = P(M_j(k)|Z_{H}^{1:k})
$$

$$
\mu_j^H(k) = p(z_H(k)|M_j(k), Z_{H}^{1:k-1}) P(M_j(k)|Z_{H}^{1:k-1})
$$

$$
\mu_j^H(k) = \frac{\Lambda_j^H(k) \bar{c}_j}{\sum_{i=1}^N \Lambda_i^H(k) \bar{c}_i}, \quad j = 1, 2, ..., N \quad (3.11)
$$

### 3.5.2 Eye-gaze Filter

The eye-gaze VS-IMM filter has a similar form to the hand motion filter described in Subsection 3.5.1.

**Interaction/Mixing:** At the beginning of each iteration, the initial conditions of the eye-gaze filter (state estimate $\hat{x}_E^{0j}(k-1|k-1)$ and covariance $\hat{P}_E^{0j}(k-1|k-1)$) are adjusted by mixing the filter outputs from the previous iteration according to

$$
\hat{x}_E^{0j}(k-1|k-1) = \sum_{i=1}^N \hat{x}_E^i(k-1|k-1)
$$

$$
\times \mu_{i|j}^F(k-1|k-1), j = 1, .., N \quad (3.12)
$$

$$
\hat{P}_E^{0j}(k-1|k-1) = \sum_{i=1}^N \mu_{i|j}^F(k-1|k-1) \hat{P}_E^i(k-1|k-1)
$$

$$
+ [\hat{x}_E^i(k-1|k-1) - \hat{x}_E^{0j}(k-1|k-1)]
$$

$$
\times [\hat{x}_E^i(k-1|k-1) - \hat{x}_E^{0j}(k-1|k-1)]^T
$$

$$
j = 1, .., N \quad (3.13)
$$

**Model Matched Filtering:** Once the initial conditions $\hat{x}_E^{0j}(k-1|k-1)$ and
\( \hat{P}_{E}^{0j}(k-1|k-1) \) are available for each filter, the state estimate and its covariance for each model are computed using the EKFs matched to the models. Along with the state estimates and the corresponding covariances, the likelihood functions \( \Lambda_{j}^{E}(k) \) are computed using the mixed initial condition (3.12) and the corresponding covariance (3.13). The likelihood \( \Lambda_{j}^{E}(k) \) is given by

\[
\Lambda_{j}^{E}(k) = p(z_{E}(k)|M_{j}(k), Z_{E}^{1:k-1})
\]

\[
\Lambda_{j}^{E}(k) = \mathcal{N}(z_{E}(k); \hat{z}_{E}^{j}(k|k-1; \hat{x}_{E}^{0j}(k-1|k-1)),
\]

\[
S_{E}^{j}(k; \hat{P}_{E}^{0j}(k-1|k-1)), \quad j = 1, \ldots, N
\]

(3.14)

where \( S_{E}^{j}(k; \hat{P}_{E}^{0j}(k-1|k-1)) \) is the innovation covariance and \( \hat{z}_{E}^{j}(k|k-1; \hat{x}_{E}^{0j}(k-1|k-1)) \) is the \( j \)th filter’s predicted measurement at time \( t \).

**Model Probability Update**: After the likelihood functions of the models \( \Lambda_{j}^{E}(k) \) are available, the model posterior probabilities \( \mu_{j}^{E}(k) \) are calculated as follows

\[
\mu_{j}^{E}(k) = P(g_{j}|Z_{E}^{1:k}) = P(M_{j}(k)|Z_{E}^{1:k})
\]

\[
\mu_{j}^{E}(k) = p(z_{E}(k)|M_{j}(k), Z_{E}^{1:k-1})P(M_{j}(k)|Z_{E}^{1:k-1})
\]

\[
\mu_{j}^{H}(k) = \frac{\Lambda_{j}^{E}(k)\bar{c}_{j}}{\sum_{i=1}^{N} \Lambda_{i}^{E}(k)\bar{c}_{i}}, \quad j = 1, 2, \ldots, N
\]

(3.15)

### 3.5.3 Determination of Human Intention

Once posterior model probabilities are available from both filters, they are fused using Equation 2.8. The goal location estimate \( \hat{g}(k) \) is then given by Equation 2.14. Figure 3.2 summarizes the gaze and motion fusion algorithm in the form of a block diagram.
Figure 3.2: This diagram summarizes the HIEVS algorithm.
3.6 Discussion

While the assumptions made in Section 3.4 may seem somewhat unreasonable if considering each filter separately, when considering the fusion of information from both filters running in parallel these assumptions can be easily justified. Consider case when the human is performing a visual search for their next goal location. If the region $R_{vs}$ is not empty, then it is most likely that the next goal is within this region or soon will be. While the set of objects which lie within the region $R_{vs}$ may change rapidly during the search, when the goal is found, it will consistently stay within this region. In this case, the parameter $\alpha$ in Equation 2.8 can be tuned based on how difficult it will be to find their next goal in a visual search. While $R_{vs}$ is not empty, the region $R_{ws}$ could be either empty or non-empty. If it is empty, then no models will be active in $M^H_a$ and the hand data will be noninformative, and the intention inference will depend solely on the gaze data. If it is non-empty then again, there are two cases. The next goal is inside the region $R_{ws}$, or it is not. If it is within the region, the wrist position will be approaching the goal object and model probability associated with the true goal will increase. If it is not, then the human will be moving their entire body closer to the goal until eventually it is inside the region $R_{ws}$. In this case, $\beta$ can be tuned based on the proximity of the object and the general speed of the human during the operation.

If $R_{vs}$ is empty, they are likely performing a visual search and they simply are looking the wrong place, in which case $R_{ws}$ is unlikely to stay empty for long. On the other hand, they may just be distracted. Either way, the gaze information during this time is noninformative and the model set $M^E_a$ is empty. If $R_{ws}$ is non-empty, the algorithm will rely entirely on the hand data. This is likely to happen towards the
end of a reaching motion wherein the current goal is close to being grasped and the human preemptively begins to search for the next object. If $R_{\text{ws}}$ is also empty, there should be a user specified safety measure in place for an uncertain case, however, under normal conditions this period would not last long.

### 3.7 Experiment

An experiment which simulates an assembly task in a large warehouse setting is designed to verify the utility the proposed method. A Microsoft Kinect sensor is placed such that a large workstation is fully visible. Within the workstation, 18 tools corresponding to $N = 18$ models are placed arbitrarily but at coordinates which are known to the algorithm a priori with respect to the Kinect reference frame. In order to model the assembly of components, the human chooses and reaches for any two objects which are not adjacent to one another in sequence. During this process, the Kinect sensor records 3D skeletal tracking data and RGB images. The RGB image feed is then passed through a deep network architecture [3] to predict the 3D gaze point. The an example of the output of the network is shown in Figure 3.3. The algorithm is run using MATLAB 2019b on an HP ELITEDESK with 8GB RAM and an AMD-PRO 3500 Mhz processor with 4 cores.

The results show that the HIEVS algorithm successfully tracks hand and gaze point motion and can correctly predict the human’s intention in both stages of the sequence. The gaze point and hand motion tracked by HIEVS are shown in Figures 3.4 and 3.5, respectively. Figure 3.6 shows the evolution of the model probabilities matched to each of the $N = 18$ goal locations. It can be seen in Figure 3.4 that the
subject’s gaze is fixated on the first goal location, corresponding to \( M_7 \), from time \( t = 0 \)s until about \( t = 1.8 \)s at which point it begins to shift to the second goal location. The saccade to the second goal, \( M_1 \), takes about 0.2s and the gaze remains fixated on this point for the remainder of the trial. Figure 3.5 shows that the hand does not start moving towards \( M_1 \) until \( t = 2.6 \)s, nearly a full second after the gaze has shifted. This occurrence is accounted for in the fusion of the posterior model probabilities in equation 2.8 by the parameters \( \alpha \) and \( \beta \) which were set to be 0.5 and 1, respectively. This means that, at first, both \( \mu^E \) and \( \mu^H \) are weighed equally, but as time goes on, \( \mu^H \) begins to hold more weight. This is observed in Figure 3.6 where the dotted red line dips. New models are activated at \( t = 1.8 \)s because the gaze point shifts. However, because the hand stays in the same location, and its weight is increasing over time, the associated model probability continues increasing. When the hand location shifts around 2.8s, the algorithm quickly recognizes a change in intention, predicting \( M_1 \) to be the most likely model at about 3.2s, which is 0.7s before the hand reaches the associated goal location, \( g_1 \).
Figure 3.4: The human gaze point tracked by the HIEVS algorithm is shown in dotted red. The measurements, in solid blue, are acquired by passing the RGB images collected by the Kinect sensor through a deep network for gaze point prediction.
Figure 3.5: Human hand motion tracked by the HIEVS algorithm is shown in dotted red. The hand motion data is acquired via the Kinect sensor’s skeletal tracking feature.
**Figure 3.6:** The evolution of the fused model probabilities associated with the $N$ models. The vertical dotted black line denotes the change in true intention.
Chapter 4

Conclusion and Future Work

The work proposed in this thesis provides a step towards seamless human-robot collaboration by contributing novel methods of human intention inference. While many of the existing human intention estimation schemes in the literature utilize only a single modality of information, the work proposed in this thesis uses the fusion of both gaze and motion data to predict human reaching tasks. In Chapter 2, an algorithm is proposed which uses a Kalman filter to track the human’s gaze point and compute corresponding model probabilities while a hand tracking IMM is used in parallel to generate predictions of the state of the human hand and corresponding model probabilities. The two model probabilities are fused before the mixing stage of the IMM on each iteration. Experimental results show the capability of this algorithm to predict human intention before the action is fully carried out. Chapter 3 proposes an algorithm names HIEVS which allows for scalability with additional models. The algorithm uses two VS-IMM filters in parallel to generate hand state estimates, gaze state estimates, and model probabilities. The model probabilities are then fused at
the end of each iteration using the same fusion equation.

Because the HIEVS algorithm allows for a large number of models to be considered in the total model set, moving forward, object detection using deep neural networks can be implemented to add additional models online. By identifying new objects of interest within the workplace and determining their location in 3D space, additional filters associated with the newly determined candidate goal locations can be added on-the-fly.
Appendix A

Additional Derivations

A.1 Learning Contracting Human Dynamics

In this section, the method used to ensure the learned human dynamics $f_i^H$ converge to goal location $g_i$ regardless of the initial condition, presented in [19], is described. Recall that in this thesis, human intention is defined as a motion profile which converges to a single goal location. Thus, it is required that each EKF associated with the $i^{th}$ model makes state predictions which tend towards the $i^{th}$ goal location.

Consider a state variable $x(t) \in \mathbb{R}^n$ and its derivative $\dot{x}(t)$. Let $\{D_i\}_{i=1}^{N_D}$ be a set of $N_D$ expert demonstrations where $\{x(t)\}_{t=0}^{T}$ and $\{\dot{x}(t)\}_{t=0}^{T}$ are recorded from time $t = 0$ to time $t = T$. Note that these demonstrations represent reaching motions toward various goal locations when learning functions $f_i^H$. The collected trajectories in $D_i$ are each translated such that they converge to the origin. Let the translated demonstrations be solutions to the underlying dynamical system governed by the first
order differential equation
\[ \dot{x}(t) = f(x(t)) + w(t) \] (A.1)

where \( f : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is a nonlinear, continuously differentiable function and \( w \sim \mathcal{N}(0, Q_e) \) is a zero mean Gaussian process noise with covariance \( Q_e \). Because each of the translated demonstrations converge to the origin, the system defined in (A.1) could be seen as a globally contracting system.

The function \( f(\cdot) \) is modeled using a neural network of the form
\[ f(x(t)) = W^T \sigma(U^T s(t)) + \epsilon(s(t)) \] (A.2)

where \( s(t) = [x(t), 1]^T \in \mathbb{R}^{n+1} \) is the input vector to the NN, \( U \in \mathbb{R}^{(n+1) \times n_h} \) and \( W \in \mathbb{R}^{n_h \times n} \) are the bounded constant weight matrices, \( \epsilon(s(t)) \in \mathbb{R}^n \) is the function reconstruction error that goes to zero after the NN is fully trained, \( n_h \) is the number of neurons in the hidden layer of the NN, \( \sigma(U^T s(t)) = [\frac{1}{1+\exp(-(U^T s(t))_1)}, \ldots, \frac{1}{1+\exp(-(U^T s(t))_{n_h})}] \) is the vector-sigmoid activation function and \((U^T s(t))_i\) is the \( i \)th element of the vector \((U^T s(t))\).

In order to train a contracting NN, the constrained optimization problem to be solved is given by
\[ \{\hat{W}, \hat{U}\} = \arg \min_{W,U} \{\alpha E_D + \beta E_W\} \] (A.3)

such that
\[ \frac{\partial f^T}{\partial x} M + M \frac{\partial f}{\partial x} \leq -\gamma M, M0 \] (A.4)

where \( E_D = \sum_{i=1}^{D} [y_i - a_i]^T [y_i - a_i], y_i \in \mathbb{R}^n \), and \( a_i \in \mathbb{R}^n \) represent the target and the network’s output of the \( i \)th demonstration, \( E_W \) is the sum of the squares of the NN weights, \( \alpha, \beta \in \mathbb{R} \) are scalar parameters of regularization, \( \gamma \in \mathbb{R} \) is a strictly
positive constant, \( M \in \mathbb{R}^{n \times n} \) represents a constant positive symmetric matrix, and the Jacobian is given by

\[
\frac{\partial f^T}{\partial x} = W^T \frac{\partial \sigma(U^T s)}{\partial x} = W^T [\Sigma'(U^T s)] U_x^T
\] (A.5)

where for any \( b \in \mathbb{R}^p \), \( \Sigma'(b) \in \mathbb{R}^{n \times n} \) is a diagonal matrix given by

\[
\Sigma'(b) = \text{diag}(\sigma(b_1)(1 - \sigma(b_1)),...,\sigma(b_p)(1 - \sigma(b_p)))
\] (A.6)

and \( U_x \in \mathbb{R}^{n \times n} \) is a sub-matrix of \( U \) formed by taking the first \( n \) rows of \( U \).

Using this method, only two NNs need to be trained, i.e. one for the reaching motion dynamics and one for the gaze-point dynamics. Functions \( f_i^H \) can be obtained by linearly translating the solutions to the dynamical system in (A.1) learned using hand motion data to the \( i^{th} \) goal location.
Bibliography


