Monash University

Department of Electrical and Computer Systems Engineering

Final Year Project

ELLIPTICAL HANDOVER REACHING MOTIONS: Towards Generating Humanlike Human-Robot Handovers

Supervisor: Dr. Wesley Chan, Prof. Elizabeth Croft

Student: Mr. Tin Tran

Student ID: 28603109

E-Mail: ttra0024@student.monash.edu

Submission Date: June 2021

SIGNIFICANT CONTRIBUTION

- Validated and compared existing models for reaching motion in handover, including the recently proposed elliptical motion model, the popular minimum jerk trajectory, and its variation: decoupled minimum jerk trajectory.

- Formulated a trajectory generator using the elliptical motion model based on observations from the dataset.

- Implement trajectory generators (ellipse, minimum jerk and decoupled minimum jerk) on robotic arm Panda Franka Emika to perform handover.

- Designed user studies to evaluate the performance of different trajectory generators based on participants' perception, such as the feeling of safety, trust, or fluency. The Monash Board of Ethics has approved this study.

ELLIPTICAL HANDOVER REACHING MOTION: TOWARDS GENERATING HUMANLIKE MOTIONS FOR HUMAN-ROBOT HANDOVER

Department of Electrical and Computer Systems Engineering, Monash University

Tin Tran
Supervisor: Dr. Wesley Chan, Prof. Elizabeth Croft

BACKGROUND

- Service robotics is more and more popular and handover is a fundamental task
- A human-like motion is preferred by human
- Reaching motion is proven to elliptical

PROJECT AIMS

- Validate elliptical model [1] on unconstrained human handover dataset [2]
- Creation of an elliptical handover motion generator
- Implementation of the created motion generator.

VALIDATION RESULT

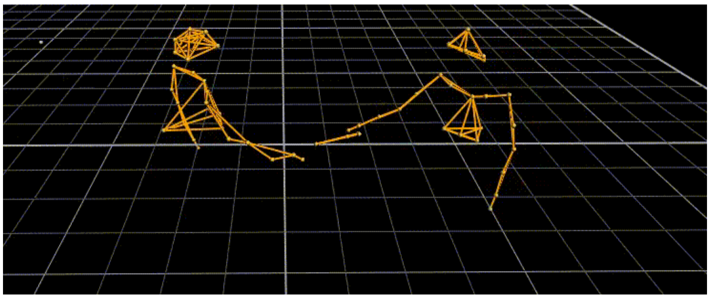


Figure 1: Motion Capture data visualized in Vicon Nexus

- In unconstrained condition, a more general form of ellipse: conic model, fits best. There is a split between hyperbolic and elliptical model.
- Out of 1195 samples from the dataset, best-fit models are distributed, along with fitting error:

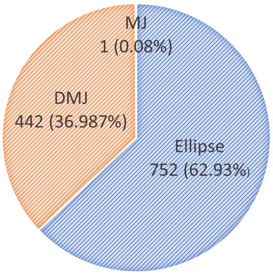


Figure 2: Fitting Result

ERROR	CONIC	DMJ	MJ
Mean (mm)	2.0	2.4	14.3
STD (mm)	3.2	3.9	12.4
MAX (mm)	34.2	34.5	121.9

Table 1: Fitting Error Statistics

- Statistical analysis reveals that there are significant differences between Conic model compared to the rest.
- **Conic model** is proved to be the **best fit.**

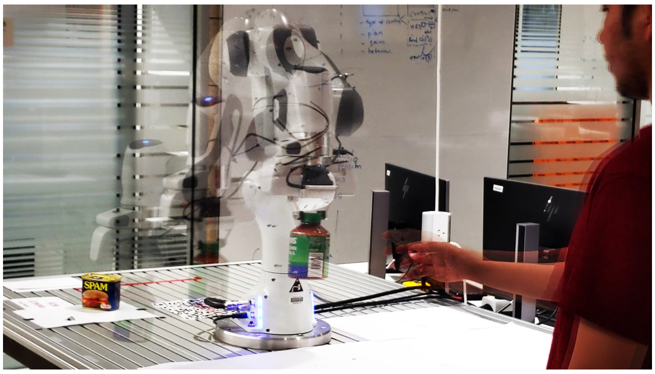


Figure 3: Robot-Human Handover using Elliptical Trajectory

TRAJECTORY GENERATOR

$$\begin{bmatrix} x(\theta) \\ y(\theta) \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} \cos(\tau) & -\sin(\tau) \\ \sin(\tau) & \cos(\tau) \end{bmatrix} \begin{bmatrix} a\cos(\theta) \\ b\sin(\theta) \end{bmatrix}$$

For the same initial and terminal condition, we found:

- Ellipse's plane's orientation are consistent.
- Arcs corresponding to handover motion are consistent.
- Major and Minor axes' length's ratio are consistent.

An ellipse is fully defined in space for arbitrary object-pickup point and handover point with these information.

The trajectory can be time parameterized with a fitted sigmoid function .

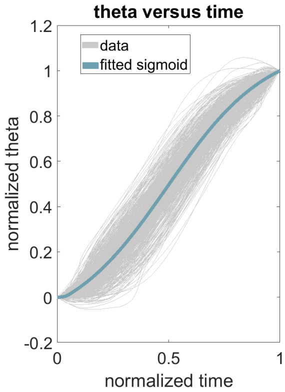


Figure 4: Fitting a Sigmoid

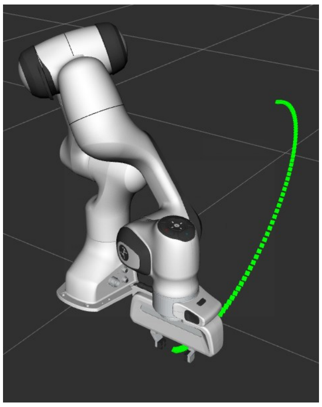


Figure 5: Elliptical Trajectory in Simulation

REFERENCE

[1] W. P. Chan, M. K. X. J. Pan, E. A. Croft, and M. Inaba, "Anaffordance and distance minimization based method for computingobject orientations for robot human handovers" IJSR, vol. 12, no. 1,p. 143–162, 2020

[2] Sheikholeslami, G. Lee, J. Hart, S. Srinivasa, and E. Croft, "Astudy of reaching motions for collaborative human-robot interaction,"in ISER, 2018

[3] M. Huber, H. Radrich, C. Wendt, M. Rickert, A. Knoll, T. Brandt,and S. Glasauer, "Evaluation of a novel biologically inspired trajectorygenerator in human-robot interaction," in RO-MAN, 2009, p. 639–644.

EXECUTIVE SUMMARY

This document is a Final Year Project report on **generating humanlike motions for a robotic arm, in Human-Robot handover scenario, where the robot is the giver**. An existing dataset on unconstrained human-human handover (where the person is allowed to move their whole body) is first processed to extract the reaching motion of the handover and fitted it against 1) the Minimum Jerk model, 2) its variation, the Decoupled Minimum Jerk model, and 3) the recently proposed Elliptical (conic) model. Results showed that **Conic model fits unconstrained human handover reaching motions best**. Furthermore, we discovered that unlike constrained, single-person reaching motions, which have been found to be mostly elliptical, **there is a split between elliptical and hyperbolic types**. We then analyzed the extracted handover reaching motions, only where the elliptical model fits best, to (successfully) generalize into an elliptical trajectory generator and implemented this trajectory generator on Panda Franka Emika robotic arm. Finally, we **designed a user study** to **evaluate the performance** of different trajectory generators based on **participants' perception (feeling of safety, trust, fluency, and working alliance)**. This user study has been **approved by Monash Board of Ethics**.

Contents

Chapter 1

Introduction

1.1 Human-Robot Handover

1.1.1 The demand

The service robotics sector has been rapidly growing in the recent decades, with much development for different applications such as homecare [1], hospital [2], manufacturing [3], and agriculture [4]. In many of these service robot applications, object handover is a fundamental task that will frequently arise. For example, handing over a screwdriver in a factory [5], handing over a TV remote at home [6], or handing over a flyer at a shopping mall [7]; thus, efficient performance of handovers is essential to the effectiveness of robots belonging to this category.

1.1.2 The challenge

While humans typically perform handovers with ease in various situations, it is still challenging for current robotic systems. Object handover is a joint-action task involving two agents, the giver and the receiver. Careful coordination between the two agents is required during the task to ensure successful transfer of the object from the giver to the receiver [8–10]. From a sensing, coordination, and safety perspective, it is much more challenging than solo object manipulation tasks such as grasping and pick-and-place, where there is no human partner involved.

1.2 Human Behaviour In Handover

1.2.1 Utilization of subtle communication channels

Humans utilize different communication channels (e.g., gaze, reaching motion, haptics, pose) during different phases of a handover task (i.e., approach phase, reaching phase, object transfer phase, retracting phase) to coordinate the different aspects of handovers (i.e., where, when, how). For example, the timing of when the giver lifts their hand and object during the approach can convey their intent to hand over the object to the receiver [11]. Gaze is used during the reaching phase to establish where object transfer is to take place [12]. During the object transfer phase, haptic channels are used to communicate when the ownership of the object is fully transferred from the giver to the receiver [13]. While human adults typically have become adept at understanding such non-explicit communications, most robots currently do not quite understand these communication modalities. Hence, to improve fluency and performance of human-robot object handovers, we need to enable robots to understand the different communication modalities used by humans during object handovers **(O1)**. For example, by teaching robots how to interpret human reaching motions, we can also enable robots to infer a human's intent to hand over an object and anticipate the location and timing of object transfer from the observed human arm motion, such that robot receivers can react in a more responsive and timely manner [14].

Many studies have investigated various communication cues used by humans for handovers and their application to human-robot handovers. Researchers have studied and compared reaching motions used in handovers [11, 15–17]. Results showed that, in general, robots employing humanlike reaching motions are preferred. The role of gaze in handovers has been investigated [12, 18], with findings indicating that the use of humanlike gaze behaviours by robots can facilitate more time-efficient handovers. Studies on dynamics in handovers have also revealed grip force control strategies and haptic-based communication used by humans for coordinating object transfer [13, 19], and demonstrated successful application to human-robot handovers [9, 10, 20]. Human body motion and kinematics leading to handovers have been studied, and classifiers have been built for predicting the occurrence of handover events [21, 22]. The role of object pose in handovers have also been studied [23, 24], and methods proposed for computing proper object configurations for handovers. These and other works have shown that enabling robots to employ humanlike behaviours improves human-robot handovers. However, human behaviours in handovers are not always well understood. Hence, towards improving human-robot handovers, there has been much research on understanding and modelling human behaviours in different aspects of handovers. **(O2)**

1.2.2 Arm reaching motions in handover

While there exist many studies on understanding the different communication channels used in the different phases of handovers, the focus of our project is on arm motions in the reaching phase of handovers. During handovers, the reaching motion of the arm not only serves to transport the object to the point of object transfer. It also serves to communicate where and when the object transfer is to take place. Much earlier studies in physiology have observed invariant properties in human reaching motions [25]. Studies have shown that human point-to-point reaching motions are roughly straight and follow a bell-shaped velocity profile [26, 27]. Hence, a person's reaching motion can provide cues to the observer about their intent and the timing and location of the reaching endpoint (i.e., where object transfer occurs). Humans naturally and subconsciously exhibit and utilize such cues for coordinating handovers. Thus, to enable fluent human-robot handovers, one widely adopted approach is to enable human-like reaching motions for robots [15, 16, 28] **(O3)**. Various models have been proposed to describe human handover reaching motions, and applied to human-robot handovers. However, there has not been empirical validations of such models to show whether these models fit natural human handover motions well - this has largely remained an assumption. Hence, to improve human-robot handovers by enabling the generation of more humanlike reaching motions, the first half of this project focuses on validating and comparing proposed models with a dataset of observed natural human handover motions.

1.3 Proposed Robot Behaviour in Handover

Having a validated model of human handover motion will allow us to implement more humanlike motions towards enabling more fluent human-robot cooperations. Furthermore, by employing humanlike reaching motions, the robot can make its action more legible and predictable to humans [29]. Legibility refers to making the robot's intent to handover the object more apparent, and predictability refers to making the robot's reaching endpoint more easily anticipated, which can potentially improve subjective metrics for human-robot handover such as the feeling of safety, trust, fluency, and working alliance.

1.4 Objectives of Project

This Final Year Project involves achieving Objective 2 **(O2)** and Objective 3 **(O3)**. Unfortunately, the scope of this project does not include Objective 1 **(O1)**. However, a parallel research conducted by Sara Sheikholeslami, with our collaboration, is tackling this exact problem.

1.5 Overview of this documents

The organization of this report is as follows: Related literature is reviewed in Chapter 2 and the way we process data is described in Chapter 3. The models fitting method are described in Chapter 4 and the fitting results are reported in Chapter 5. A method to formulate a trajectory generator is then proposed in Chapter 6, and the user study design is presented in Chapter 7. We discuss the results and future work in Chapter 8 before concluding in Chapter 9.

Chapter 2

Related Work

Human handovers typically happen very fast (within ~1.2s according to our dataset). Thus, to enable online generation of handover trajectories, a computationally fast model is required. Although more advanced and complex models such as those drawn from biomechanics or Inverse Optimal Control for describing human motions exist, they are less suitable for the target application of real-time human-robot handovers. In this section, we describe existing models that have been proposed for and/or applied to human-robot handovers.

2.1 Reaching Motion Models

As mentioned in section 1.2.2, early studies have shown that human reaching motions exhibit invariant properties. The Minimum Jerk trajectory model has been formulated for describing such observed human reaching motions in the 1980s[30, 31]. Since then, this model has been widely accepted and applied to robot reaching motions in various human-robot interaction tasks, including object handovers, and continues to be used for modelling human handover motions in recent works [32, 33]. Early on, velocity profiles resembling the Minimum Jerk trajectory were applied to 1D tabletop human-robot handovers [15]. Compared to typical industrial robot trajectories, a minimum-jerk-like trajectory allowed the robot to be perceived as more careful, pleasant, and skilled. Later, the Minimum Jerk trajectory was applied to 3D handover tasks seated at a table [16]. Similarly, a Minimum Jerk trajectory allowed the robot to be perceived as safer with a shorter reaction time from human receivers. While the original applications of the Minimum Jerk model assumed that the path, like human reaching motions, is, in general, approximately straight [26, 27], it has later been shown that reaching motions in handovers are curved [28]. As a result, a *Decoupled Minimum Jerk* trajectory model was proposed, which decouples the 3D reaching motion into two Minimum Jerk trajectories - one in the z direction, and one in the orthogonal xy plane [28]. This results in a curved trajectory but still has the characteristic that the ending portion of the trajectory is straight. While existing studies demonstrated the benefits of employing human-like motions for human-robot handovers, they have not explicitly fitted the proposed models to experimental data to evaluate how well these models describe human handover motions.

Recently, a new *Elliptical model* has been proposed for human reaching motions. This model has been experimentally shown to accurately fit human reaching motions in single-person reaching tasks [29]. In this study, the experimenter investigated a seated, tabletop pick-and-place task. It was suggested that the Elliptical model could potentially be applied to reaching motions in human-robot interaction tasks such as handovers. However, it has not yet been shown that human reaching motions in the joint action task of handover also follow the proposed Elliptical model.

2.2 Constrained versus Unconstrained Handover

Literature shows that movement planning and control for single-person reaching motions vary with task context [34]. It has also been shown that there is online coordination and adaptation between the giver's and receiver's actions during handover tasks [13, 14]. Hence, reaching motions in single-person tasks and joint action tasks such as handovers may be different. Furthermore, existing studies have been primarily limited to constrained tasks, where participants performed the task seated, over a tabletop handling one generically shaped object [15, 16, 28, 29]. This is quite different to everyday handovers that service robots will need to perform. Thus, we aim first to verify if human reaching motions in unconstrained handovers, handing over everyday objects, fit existing proposed models, and evaluate which model fits best.

This report will discuss in detail the three implemented models in Chapter 4.

Chapter 3

Dataset Processing

3.1 Human Handover Dataset

This project analyzed the publicly available Handover Orientation and Motion Capture Dataset[1] [35], which contains 1200 unconstrained human-human handover trajectories performed by twenty participants recorded by a Vicon motion capture system. In the dataset, twenty common objects (Fig. 3.1) were used, and each handover started with the giver and receiver standing facing each other, and one of the objects placed on a table left, right, or behind the giver. Fig. 3.2 shows the data collection setup. The giver picked up the object and handed it over to the receiver using unconstrained motion (i.e., they were allowed to move their whole body). The only constraint in this dataset is that the giver can only use their right hand to perform handover for consistency. There are also three modes the participants have to follow for each object: A - natural handover, B - giver centered handover (focus on the comfort of the giver), and C - receiver centered handover (focus on the comfort of the receiver). Additional details of the data collection procedure can be found in [35].

[1] https://bridges.monash.edu/articles/Handover_Orientation_and_Motion_Capture_Dataset/8287799

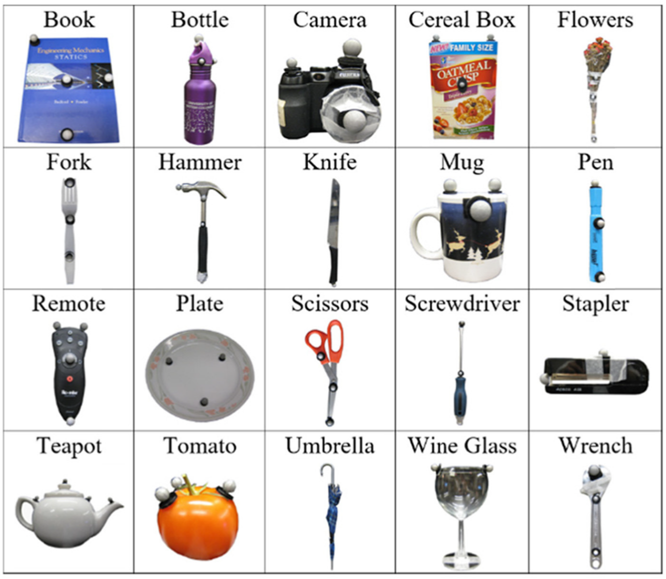


FIGURE 3.1: The twenty common everyday objects used for handovers in the dataset. (Image from [35].)

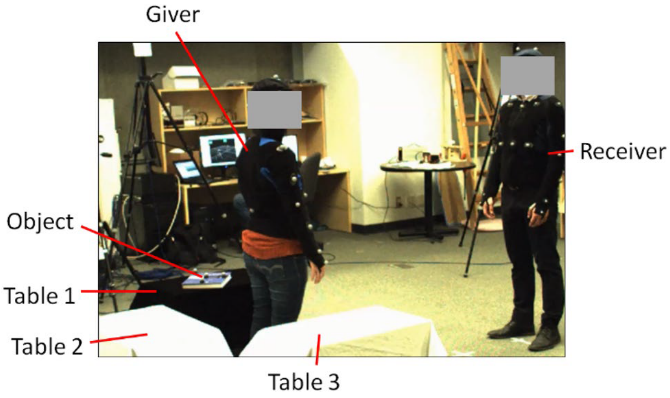


FIGURE 3.2: Data collection setup for collecting human handover motions in the dataset. (Image from [35].)

3.2 Handover Segmentation

For pre-processing, few missing data points of each sample (if there are any) are interpolated using MATLAB function `interp1` with `spline` option. Performing Spectral

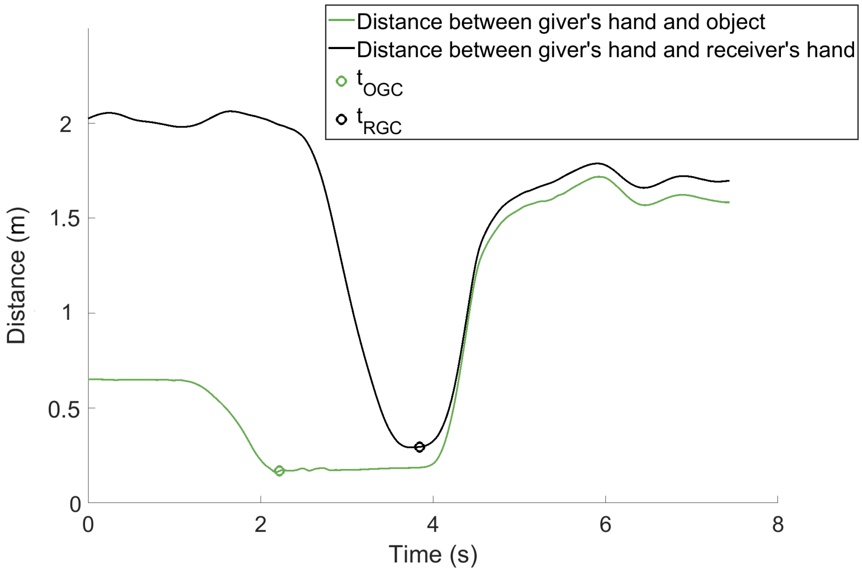


Figure 3.3: Plot of distance between giver's hand and object, and distance between giver's hand and receiver's hand in an example handover trial.

analysis [2] on the dataset reveals most signal power reside below 7Hz, so we filtered all handover trajectories using a 10 Hz low pass Butterworth filter. A median filter with windows length = 3 is ten applied to remove any irregular jumps.

To segment the handover reaching motion, we first identified the instance when the giver first touches the object, t_{OGC}, and the instance when object transfer occurs, t_{RGC}. These instances are found by identifying the point in time when the distance between the giver's hand and object reaches a minimum, and when the distance between the giver's hand and receiver's hand reaches a minimum, respectively. Fig. 3.3 shows an example from a typical trial. The end of the reaching motion, t_{end}, is then simply t_{RGC}.

Determining the start of the handover reaching motion, t_{start}, however, requires an extra step. Inspecting the data, we observed two common giver tendencies: Case 1) reaching directly towards receiver after object pick up (73.97%), Case 2) first bringing the object closer to themselves before reaching towards the receiver (26.03%). Fig. 3.5a, 3.5b shows example speed profiles demonstrating the two tendencies. In the latter case, the giver

[2] using `ftt` from MATLAB to perform Fourier Transform on the signal, and computing the power of Discrete Fourier Transform as guided on the official document provided by MATLAB https://au.mathworks.com/help/matlab/math/basic-spectral-analysis.html

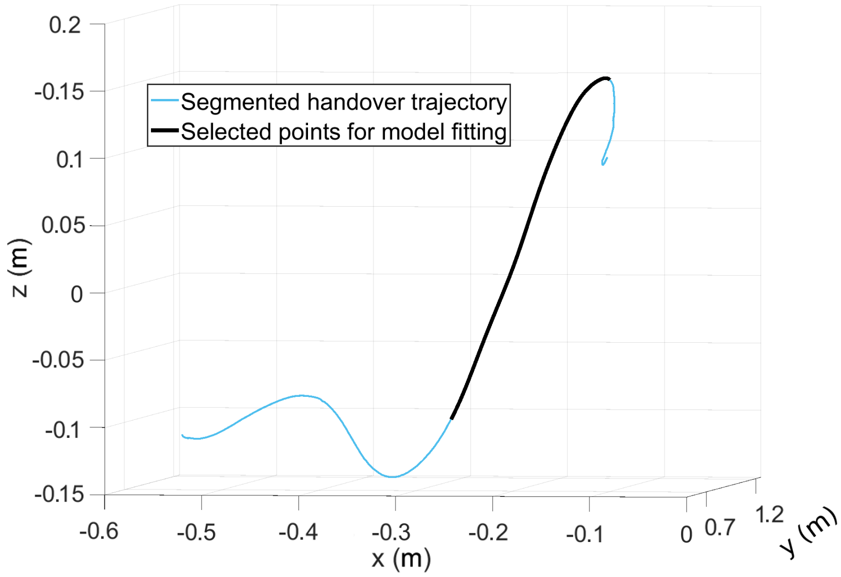
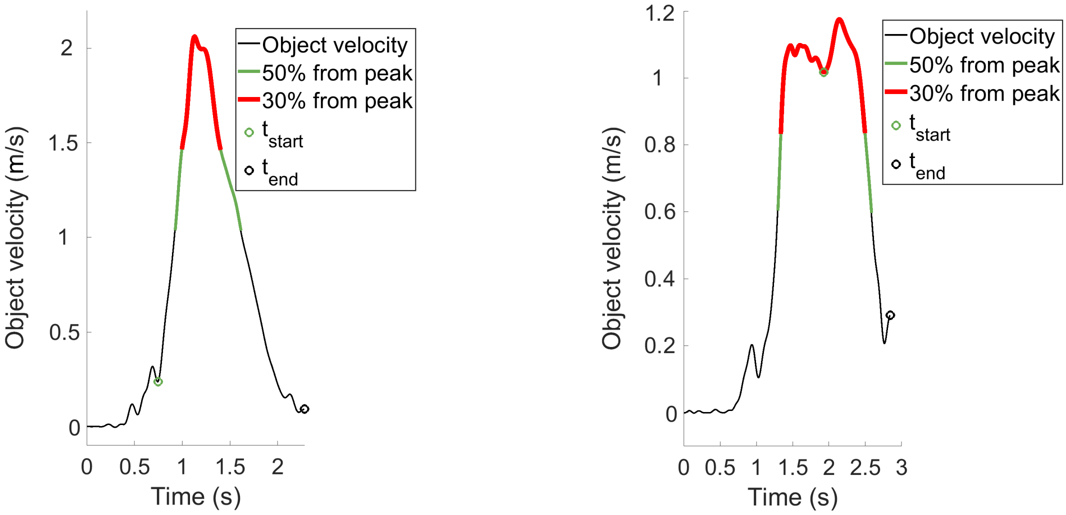


Figure 3.4: Segmented handover reaching trajectory)

slows down slightly as they bring the object closer to themselves, manifesting a trough. To further validate this description, multiple live samples were inspected using Vicon Nexus software (software to visualize and playback all the recorded data by Vicon), where the handover is classified as Case 2. The trough in velocity indeed corresponds to the described action. Hence, we detect this by inspecting if the region where the object's speed is >70% peak speed (Fig. 3.5 red portion) contains any local minimum. In Case 2, the start of the handover reaching motion, t_{start}, is determined to be this local minimum, while in Case 1, t_{start} is determined to be the last local minimum that is <50% peak speed (Fig. 3.5 black portion).

Since this handover dataset is unconstrained, we sometimes find noisy data points near the beginning and the end of the segmented handover reaching motion, mostly due to artefacts of different motion blending together, i.e. object picking motion, getting ready motion, fine adjustment towards the end, etc. To filter this out, when fitting each trajectory model, we included only points that lie within 3 standard deviations of the best fit plane. The best fit plane is estimated using Δ% of the segment, terminated at the point that is δ mm below the highest point in the segment. This is because towards the end, the fine adjustment of the giver to meet the receiver's hand often diverges from the reaching plane since it highly depends on the mode of the handover and how good the mutual understanding of the 2 participants on the object transfer point was. Hence, we



(A)

(B)

Figure 3.5: Typical speed profile of an object. a - Giver immediately hands an object over after picking it up. b - Giver brings the object closer to themselves first after picking up the object before handing over.

terminted the initial guess of best fitted plane before the end points of the segmentation so far. The algorithm can be described as:

Algorithm 1: Final adjustment to extract handover reaching motion

```
$T$ is a Cartesian point, index start at 1, i.e. $T[1]$ is x, $T[2]$ is y, $T[3]$ is z;
$\mathbf{T}$ is the segmented reaching motion so far. $\mathbf{T} = \{T_j\}$, $j = 1 \cdots M, \mathbf{T} \in \mathbb{R}^{3\times N}$;
$N = \Delta\% \times M$;
$T_A \in \mathbf{T}$, such that $T_A[3] \geq T_j[3]$, $j = 1 \cdots M$;
$T_B \in \mathbf{T}$, such that $T_B[3] + \delta < T_A[3]$, $T_{B+1}[3] + \delta > T_A[3]$, $B < A$;
$\mathbf{Q} = \{T_i\}$, where $i \in \{B - N, B - N + 1, B - N + 2, \cdots, B\}$;
$\mathcal{P}$ is the plane that fit cluster $\mathbf{Q}$ (details in section 4.2.1);
while $\mathcal{P}$ does not change do
    $\mathbf{Q} = \{T_k\} \subset \mathbf{T}$, where $T_k$ is within 3 stardard deviation of distance to plane $\mathcal{P}$;
    $\mathcal{P}$ = best fit plane for $\mathbf{Q}$;
end
return $\mathbf{Q}$
```

We empirically set Δ=20 and δ=0.005 to capture as many coplanar points within the handover trajectory as possible while excluding other data points belonging to arm motion before and after the handover trajectory. Fig. 3.4 shows the segmented handover trajectory and data points included for model fitting from one typical example trial.

Chapter 4

Models Fitting

4.1 Models

4.1.1 Elliptical (Conic) Model

Recently, it has been empirically shown that the trajectory of human reaching motions in pick-and-place tasks is curved in 3D space but planar, and that an elliptical curve achieves a good fit to the path drawn out by the projection of the trajectory on the best-fit plane [29]. A general conic section on a given plane can be expressed as:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{4.1}$$

Once we found the best fit coefficients for a particular path in space, we can classify the conic section by its discriminant [1]: Ellipse if $B^2 - 4AC < 0$, Parabola [2] if $B^2 - 4AC = 0$, Hyperbola if $B^2 - 4AC > 0$, or Circle if $A = C$ and $B = 0$ [36].

[1] Since we are looking at curves, we will not consider the degenerate case where $A = B = C = 0$, or the case where trivial solution is the only solution, such as when $B = 0$, and $D^2/4A + E^2/4C - F > 0$

[2] In practice, if $B^2 - 4AC <$ `1e-12`, $B^2 - 4AC$ is considered to be equal to 0. However, samples that are identified as parabola with this threshold are fitted with hyperbolic model. Condition number for these calculation are not high because $B^2 - 4AC >$ `1e-15` for all samples

If the conic section is classified as an ellipse, we can express it in parametric form [37] as

$$\begin{bmatrix} x(\theta) \\ y(\theta) \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} \cos(\tau) & -\sin(\tau) \\ \sin(\tau) & \cos(\tau) \end{bmatrix} \begin{bmatrix} a\cos(\theta) \\ b\sin(\theta) \end{bmatrix} \tag{4.2}$$

where x_c, y_c is the ellipse center, τ the inclination angle, and a, b the semi-major and semi-minor axes. Using this form, we can generate a trajectory by specifying the dependence of θ on time. Notice that in both forms, we only need five parameters to fully define an ellipse as this will be proven useful later when we try to formulate a trajectory generator in Section 6.3.

If the conic section is classified as a hyperbola, we can also express it in parametric form as

$$\begin{bmatrix} x(\theta) \\ y(\theta) \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} \cos(\tau) & -\sin(\tau) \\ \sin(\tau) & \cos(\tau) \end{bmatrix} \begin{bmatrix} a\sec(\theta) \\ b\tan(\theta) \end{bmatrix} \tag{4.3}$$

4.1.2 Minimum Jerk Model

The Minimum Jerk model hypothesizes that human arm motion between two points in space minimizes the jerk over the entire path [30]. Given position over time $r(t)$, jerk is the third derivative of position, $\dddot{r}(t)$. The smoothness of the path can be measured by

$$S = \int_0^{t_f} \dddot{r}(t)^2 dt \tag{4.4}$$

where t_f is the duration of the motion. The Minimum Jerk trajectory is the trajectory $r(t)$ that minimizes the quantity S given by Eq. 4.4. The solution [30] takes on the form of a fifth degree polynomial:

$$r(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \tag{4.5}$$

where the coefficients a_i are determined by the boundary conditions, i.e., the position, velocity, and acceleration at start and end points of the trajectory (further details in Appenix A). The Minimum Jerk trajectory assumes a straight line path between the start point and the end point.

4.1.3 Decoupled Minimum Jerk Motion Model

Literature has shown that if we decouple the z axis motion from the xy plane motion by specifying two Minimum Jerk trajectories with different durations for the two components, we obtain a curved trajectory that more closely resembles human handover reaching motions [28]. This Decoupled Minimum Jerk trajectory is described by

$$r_z(t) = a_{0z} + a_{1z}t + a_{2z}t^2 + a_{3z}t^3 + a_{4z}t^4 + a_{5z}t^5, \tag{4.6}$$

$$r_{xy}(t) = a_{0xy} + a_{1xy}t + a_{2xy}t^2 + a_{3xy}t^3 + a_{4xy}t^4 + a_{5xy}t^5, \tag{4.7}$$

where $r_z(t)$ is the trajectory in the z direction, with duration t_z, and $r_{xy}(t)$ is the trajectory in the xy plane, with duration t_{xy},. The coefficients a_{iz} and a_{ixy} are determined by their boundary conditions and corresponding duration (further details in Appenix A). The Decoupled Minimum Jerk trajectory results in a curved path in 3D space, residing in a plane orthogonal to the xy plane.

4.2 Fitting Procedure

4.2.1 Plane Fitting

If we have a set of n data points in Cartesian space, we can construct the following matrix

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \end{bmatrix}_{3\times n}$$

Which can then be used to construct the covariance matrix

$$\mathbf{\Sigma} = \frac{(\mathbf{X} - \mathbf{X_{centroid}})(\mathbf{X} - \mathbf{X_{centroid}})^T}{n}$$

Performing Principle Component Analysis (PCA) on matrix $\mathbf{X}$, essentially finding the eigenvectors of $\mathbf{\Sigma}$, will give us three eigenvectors denoting the directions in which the data spreads, and their corresponding eigenvalue denoting how much the data is spreading in that direction [38]. If $\mathbf{X}$ is indeed planar, then there exists an eigenvalue that is significantly smaller compared to the others. Since eigenvectors obtained from PCA are orthogonal to each other, the eigenvector corresponding to the smallest eigenvalue is the normal vector to this plane. Because a plane is fully defined by its normal vector (and the centroid), we found the best fitted plane. MATLAB conveniently offers function `pca` to compute the principal components directly.

4.2.2 Error Calculation

For each fitted model, we compute the fitting error as the average Euclidean distance from each data point to the correspondent point on the fitted model:

$$err = \frac{1}{N}\sum_{i}^{N} ||\boldsymbol{r}_{gi} - \boldsymbol{r}_i|| \tag{4.8}$$

where $\boldsymbol{r}_i$ is the i^{th} data point in reaching trajectory, and $\boldsymbol{r}_{gi}$ is the correspondent point on the fitted model. We define the correspondent point as the point on the fitted curve with the smallest Euclidean distant (closest point) from the data point. For each motion model, the correspondent point $\boldsymbol{r}_{gi}$ to each data point $\boldsymbol{r}_i$ is computed as described in section 4.2.5, 4.2.3, and 4.2.6.

Note: Decoupled Minimum Jerk and Minimum Jerk are techniques to generate trajectory with respect to time. However, we are more interested in the path fitting aspect at this stage; therefore, correspondent points for Minimum Jerk and Decoupled Minimum Jerk are found using the closest point on the path to a data point, instead of using time parameter.

4.2.3 Minimum Jerk Model

Since the Minimum Jerk trajectory path is defined as a straight line in 3D, finding the correspondent point reduces to finding the closest point on a line to a given point. This can be easily solved analytically. Let $\boldsymbol{P}_1$ be the start point, and $\boldsymbol{P}_2$ be the final point, on the reaching trajectory. The correspondent point is given by

$$\boldsymbol{r}_{gi} = \boldsymbol{P}_1 - (\boldsymbol{P}_2 - \boldsymbol{P}_1)\left(\frac{(\boldsymbol{P}_1 - \boldsymbol{r}_i)\cdot(\boldsymbol{P}_2 - \boldsymbol{P}_1)}{|\boldsymbol{P}_2 - \boldsymbol{P}_1|^2}\right) \tag{4.9}$$

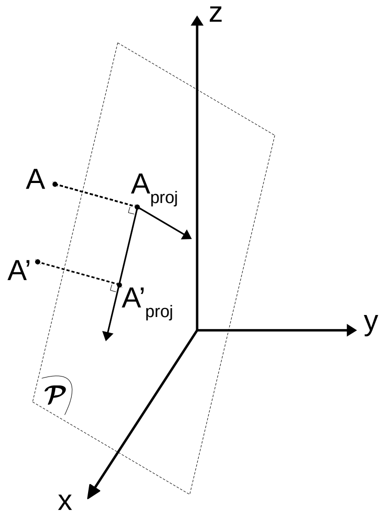


FIGURE 4.1: Projecting Data Points to plane $\mathcal{P}$. A is a data point, A_{proj} is the projection of A on $\mathcal{P}$, A'_{proj} is the correspondent point of A_{proj} (the difference is exaggerated for visualization), and A' is A'_{proj} projected back to 3D space.

,

4.2.4 Fitting 2D model to 3D data points

Once we found the best fit plane $\mathcal{P}$, to fit a 2D model, one approach would be projecting all the segmented data points onto that plane and fit a model. While this is straightforward in the "feedforward" direction, converting the 2D correspondent points back to 3D is a cumbersome process because:

- We have to store the signed distances from each data points to the plane. The sign depends on which side of the plane the data point is on.
- From figure 4.1, it is clear that we have to keep track of the arbitrary bases of plane $\mathcal{P}$ that the projected point is expressed in. Because we have to express A'_{proj} back in the original coordinate before obtaining A' (by using the signed distance and the normal vector of plane $\mathcal{P}$).

Hence, instead of this approach, the following procedures will offer a more elegant and efficient way to transform the data points for fitting:

1. Fit plane $\mathcal{P}$ to the segmented reaching motion ($\mathbf{Q}$ from Algorithm 1, $\mathbf{Q} \in \mathbb{R}^{3\times N}$), obtaining normal vector $\vec{n}$ of $\mathcal{P}$.

2. Rotate $\mathcal{P}$ such that $\mathcal{P}$ is horizontal, i.e. aligning $\vec{n}$, or $\hat{n}$ (normalized $\vec{n}$) with $\hat{k}$ (unit vector of z axis). Applying Rodrigues' rotation formula will yield rotation matrix $\mathbf{R}$ (Appedix B).

3. Left-multiply all the data points in the segmented reaching motion with $\mathbf{R}$ to rotate as described in step 2, $\mathbf{Q}_{rot} = \mathbf{RQ}$, $\mathbf{Q}_{rot} \in \mathbb{R}^{3\times N}$.

4. Remove the z component of the rotated data points, $\mathbf{Q}$, to effectively collase all points onto xy plane, obtaining $\mathbf{Q}_{2D}$

$$\mathbf{Q}_{rot} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \end{bmatrix}_{3\times n} \Rightarrow \mathbf{Q}_{2D} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix}_{2\times n}$$

5. Use $\mathbf{Q}_{2D}$ to fit different 2D models: elliptical model in section 4.2.5, decoupled minimum jerk model in section 4.2.6. Then, we obtain the matrix of correspondent points, $\mathbf{Q}_{2D}^{corr}$

6. Restore the z component (from $\mathbf{Q}_{rot}$) for each 2D correspondent point

$$\mathbf{Q}_{2D}^{corr} = \begin{bmatrix} x_{1_{corr}} & x_{2_{corr}} & \cdots & x_{n_{corr}} \\ y_{1_{corr}} & y_{2_{corr}} & \cdots & y_{n_{corr}} \end{bmatrix}_{2\times n} \Rightarrow \mathbf{Q}_{2D}^{corr} = \begin{bmatrix} x_{1_{corr}} & x_{2_{corr}} & \cdots & x_{n_{corr}} \\ y_{1_{corr}} & y_{2_{corr}} & \cdots & y_{n_{corr}} \\ z_1 & z_2 & \cdots & z_n \end{bmatrix}_{3\times n}$$

   then we left-multiply $\mathbf{Q}_{2D}^{corr}$ with $\mathbf{R}^{-1}$ to find the fitted correspondent point in 3D

$$\mathbf{Q}_{3D}^{corr} = \mathbf{R}^{-1}\mathbf{Q}_{2D}^{corr}$$

   where i^{th} column of $\mathbf{Q}_{3D}^{corr}$ is $\boldsymbol{r}_{gi}$

4.2.5 Elliptical (Conic) Model

With Elliptical model, after obtaining $\mathbf{Q}_{2D}$ from step 5 of section 4.2.4, our objective is to find the best-fit coefficients, i.e. A, B, C, D, E, F in Equation 4.1. Without loss of generality, we can divide both sides of Equation 4.1 by F, meaning:

$$\frac{A}{F}x^2 + \frac{B}{F}xy + \frac{C}{F}y^2 + \frac{D}{F}x + \frac{E}{F}y + 1 = 0 \tag{4.10}$$

Since A, B, C, D, E, F are just scalars, we can express Equation 4.10 as

$$Ax^2 + Bxy + Cy^2 + Dx + Ey = -1 \tag{4.11}$$

Equation 4.11 can be expressed in vector forms as

$$\begin{bmatrix} x^2 & xy & y^2 & x & y \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = -1 \tag{4.12}$$

With each column in $\mathbf{Q}_{2D}$, we essentially add one more constraint (linear equation) to solve for the coefficients. By stacking all the constraints, we can express our system of equations as

$$\underbrace{\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 \\ . & . & . & . & . \\ x_n^2 & x_ny_n & y_n^2 & x_n & y_n \end{bmatrix}_{n\times 5}}_{\mathbf{S}} \underbrace{\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} -1 \\ -1 \\ . \\ -1 \end{bmatrix}_{n\times 1}}_{\mathbf{b}} \tag{4.13}$$

Equation 4.13 essentially describes an overdetermined system. We can use the least-square optization method to find the approximate solution [39]. Equation 4.14 shows how to obtain the coefficients.

$$\mathbf{c} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{b} \tag{4.14}$$

With $\mathbf{c}$, we can classify the curve into ellipse, hyperbola, or parabola (section 4.1.1). However, regardless of classification, we can still determine the principle axis, their lengths, and the center [40]. To summarize, these parameters (using the same notation as Equation 4.2 and Equation 4.3, where possible) can be calculated as

$$a = \sqrt{-\Delta/(\lambda_1^2\lambda_2)} \tag{4.15}$$

$$b = \sqrt{-\Delta/(\lambda_1\lambda_2^2)} \tag{4.16}$$

$$x_c = (BE - 2CD)/(4AC - B^2) \tag{4.17}$$

$$y_c = (BD - 2AE)/(4AC - B^2) \tag{4.18}$$

$$m_1 = 2(\lambda_1 - a)/B \tag{4.19}$$

$$m_2 = 2(\lambda_2 - a)/B \tag{4.20}$$

where

- m_1, and m_2 are the slopes of the principle axis with length a, and b respectively. *Note*: $m_1 m_2 = -1$ because the principle axes are orthogonal to each other.
- λ_1, and λ_2 are the eigenvalues of $\begin{bmatrix} A & B/2 \\ B/2 & C \end{bmatrix}$
- $\Delta = \begin{vmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{vmatrix}$, with operator $|\cdot|$ denotes the determinant of a matrix.

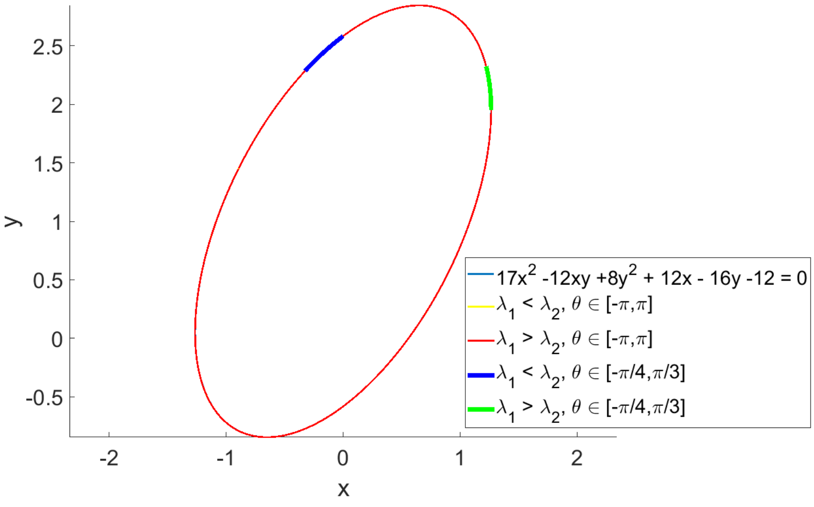
For the sake of consistency, which will also be beneficial for formulating a trajectory generator later in Section 6.3.1, we have to establish an order for λ_1 and λ_2, because a, b and τ depends on this order. Figure 4.2 demonstrates how the order of λs affects the parametric form of an ellipse and a hyperbola.

For an ellipse: The order of λ does not change the shape of the parametric form. However, different orders will yield different segments of the ellipse corresponding to the same θ segment, i.e. with $\theta = pi/3$, different λ order results in different point in space, as in Figure 4.2a. Within this project, the convention will be

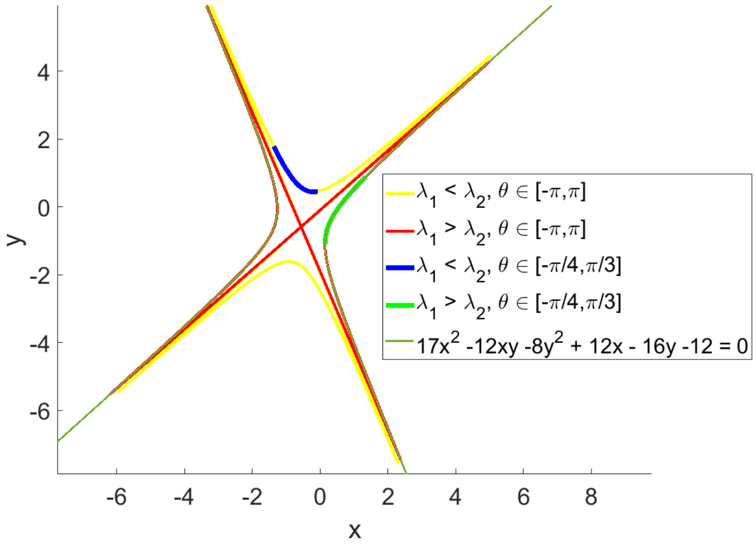
$$|\lambda_1| \leq |\lambda_2|$$

Consequently, this means $a > b$ because a is inversely proportional with λ_1. τ is then calculated as

$$\tau = \tan^{-1}(m_1), \tau \in [-\frac{\pi}{2}, \frac{\pi}{2}]$$



(A) Ellipse



(B) Hyperbola

FIGURE 4.2: Effect of the order of λ on parametric form of a conic sections.

Geometrically, this is equivalent to having an origin-centered horizontal ellipse (i.e. the longer principle axis is parallel to x-axis) rotated by τ and translated by (x_c, y_c). If we pick the other order, i.e. $|\lambda_1| \geq |\lambda_2|$, it is equivalent to having an origin-centered vertical ellipse (i.e. the longer principle axis is parallel to y-axis) rotated by $-(\pi/2 - \tau)$ and translated by (x_c, y_c).

For a hyperbola: Different λ order will yield different shape of the parametric form.

This λ order will dictate the direction of opening of the hyperbola. Hence, the solution for the order of λ is unique in this case, contrasted to the ellipse. To solve this, we can use the eccentricity of a conic section. Eccentricity, along with the lengths of the principal axes, are the only invariants of a conic section - a quantity that does not change when the conic undergoes isometric transformation [41]. For a hyperbola written in general conic section form, as described by Equation 4.1, its eccentricity can be calculated as

$$e_g = \sqrt{\frac{2\sqrt{(A-C)^2+B^2}}{-(A+C)+\sqrt{(A-C)^2+B^2}}} \tag{4.21}$$

And by definition, the eccentricity of a hyperbola in parametric form (Equation 4.3) is calculated as

$$e_p = \frac{\sqrt{a^2+b^2}}{a} \tag{4.22}$$

So we have to choose λ order such that $e_p = e_g$. τ is then calculated as

$$\tau = \tan^{-1}(m_1), \tau \in [-\frac{\pi}{2}, \frac{\pi}{2}]$$

Note: We can not apply the same strategy (using eccentricity) for an ellipse fitting because the eccentricity of an ellipse in parametric form is calculated as

$$e_p = \frac{\sqrt{c^2-d^2}}{c}$$

Where $c = \max\{a, b\}$, and $d = \min\{a, b\}$. Hence, regardless of either $|\lambda_1| < |\lambda_2|$ or $|\lambda_2| < |\lambda_1|$, e_p will always be equal to $|e_g|$. Hence, eccentricity approach is not useful.

Once we obtain the parametric form of a conic section, either ellipse or hyperbola, we can then find the correspondent point. Since the parametric form of a conic is a function of angle θ, this means that ultimately, we would want to find the corresponding θ of the model to a single data point. This can be done through optimization formulated with

$$\min_{\theta} ||\hat{\mathbf{x}}(\theta) - \mathbf{x}||_2 \tag{4.23}$$

where $\hat{\mathbf{x}}(\theta) = \begin{bmatrix} x(\theta) & y(\theta) \end{bmatrix}^T$ is a point generated by the fitted model with angle θ, $\mathbf{x}$ is the data point we try to find the correspondent of. We use `fminsearch` function provided by

MATLAB to find the solutions. Since optimization is sensitive to initial guess, Algorithm 2 describes a method to efficiently solve for all θ of a segmented handover reaching motion

Algorithm 2: Finding all correspondent θ to a segmented handover reaching motion

```
Input: $\mathbf{Q}_{2D} = \{Q_i\}$, $i = 1, 2, \cdots N$, $Q_i$ is $\mathbf{x}$ in Equation 4.23
Output: $\mathbf{\Theta} = \{\theta_i\}$, $i = 1, 2, \cdots N$; $\mathbf{Q}_{2D}^{corr} = \{Q_i^{corr}\}$, $Q_i^{corr}$ is $\hat{\mathbf{x}}$ in Equation 4.23
Function FindTheta($Q_i$, $\theta$):
    $\theta_i \leftarrow$ solution of optimization with objective 4.23, initial guess $\theta$ ;
    return $\theta_i$;
MinError $\leftarrow \infty$ ;
for $\leftarrow 1$ to 20 do
    $\theta_0 \leftarrow RAND(-\pi, \pi)$ ;
    $\theta_1 \leftarrow$ FindTheta ($Q_1$, $\theta_0$) ;
    $Q_1^{corr} \leftarrow$ parametric equation evaluated at $\theta_1$;
    for $k \leftarrow 2$ to $N$ do
        $\theta_k \leftarrow$ FindTheta ($Q_k$, $\theta_{k-1}$) ;
        $Q_k^{corr} \leftarrow$ parametric equation evaluated at $\theta_k$
    end
    TotalError $\leftarrow ||\mathbf{Q}_{2D} - \mathbf{Q}_{2D}^{corr}||_2$;
    MinError $\leftarrow$ min(MinError, TotalError);
end
$\mathbf{\Theta} \leftarrow \mathbf{\Theta}$ such that TotalError = MinError;
$\mathbf{Q}_{2D}^{corr} \leftarrow \mathbf{Q}_{2D}^{corr}$ such that TotalError = MinError;
```

In summary, to find the correspondent, first, we initialize a random guess for θ of the first point in the data point array, $\mathbf{Q}_{2D}$. Optimization is then used to find the correspondent of the first point in the data point array. Then we use that first correspondent point as the initial guess to find the second correspondent point. At this point, we iterate until we find all the correspondent points. This whole process is done 20 times with 20 random initializations, and the array of correspondents with the lowest overall error is the best-fit. Hence we obtained $\mathbf{Q}_{2D}^{corr}$ with its corresponding $\mathbf{\Theta}$

4.2.6 Decoupled Minimum Jerk Model

In decoupled minimum jerk motion model, there are 2 parameters we can tune, the duration of the motion along xy plane, t_{exy} and the duration along z axis, t_{ez}. Since the duration is finite, without loss of generality, we can normalize both the duration with the duration along xy plane. With this normalization, $t_{exy} = 1 = \text{const}$ and $t_{ez} \in (0, \infty)$. Since we are fitting a set of 2D points in xy plane, we will abuse $x(t)$ and $y(t)$ notation, which have been familiar with us in section 4.2.5 so far. This is still valid because it is effectively a change of coordinate system, where a vector in xy plane and z being the bases for Equation 4.6 and Equation 4.7, x and y are the bases for $\mathbf{Q}_{2D}$. We can then express model in this 2D plane as

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_{0xy} + a_{1xy}t + a_{2xy}t^2 + a_{3xy}t^3 + a_{4xy}t^4 + a_{5xy}t^5 \\ a_{0z} + a_{1z}t + a_{2z}t^2 + a_{3z}t^3 + a_{4z}t^4 + a_{5z}t^5, \end{bmatrix} \tag{4.24}$$

The coeffcients can be found using boundary condition of $\mathbf{Q}_{2D}$. In practice, boundary condition was solved using $\mathbf{Q}_{2D}^{extra}$. If $\mathbf{Q}_{2D} = \{Q_i\}, i = 1 \cdots N$, then $\mathbf{Q}_{2D}^{extra} = \{Q_i\}, i = -2, -1, \cdots N+1, N+2$. Heading and trailing elements will help with more accurate calculation of the initial velocity and acceleration. To solve for t_{ez} and the correspondent points, we effectively solve N optimizations inside an optimization. The outer objective can be formulated as

$$\min_{t_{ez}} ||\mathbf{Q}_{2D} - \mathbf{Q}_{2D}^{corr}||_2 \tag{4.25}$$

Where each column in $\mathbf{Q}_{2D}^{corr}$, i.e. each correspondent point generated by Decoupled Minimum Jerk Model, for a given t_{ez} (because the coefficients of DMJ in Equation 4.24 depends on t_{ez}), can be found through optimization with the inner objective

$$\min_{t} ||\hat{\mathbf{x}}(t) - \mathbf{x}||_2 \tag{4.26}$$

Again, we can find the best-fit t_{ez} and $\mathbf{Q}_{2D}^{corr}$ using `fminsearch` offered my MATLAB.

4.3 Comparison of Motion Models

To determine which model best fits unconstrained handover reaching motions, we conducted an ANOVA analysis, followed by t-tests with Bonferroni correction (alpha = 0.05) to compare the fitting error of the three models (Conic, Minimum Jerk, Decoupled Minimum Jerk). We then examined the conic fit results to determine the percentage of elliptical, parabolic, and hyperbolic curves.

Chapter 5

Fitting Results

5.1 Statistical Results

Out of the 1200 handovers in the dataset, four trials were excluded due to missing too much trajectory data, and one trial was identified as an outlier where participants handover with their left hand. Fig. 5.1 shows the three fitted models for an example trial in 3D space. Fig. 5.2 shows the fitted Conic and Decoupled Minimum Jerk models for the same example trial projected on the 2D best fit plane. Table 5.1 summarizes resulting fitting error for each model. Examining individual trials, among the 1195 fitted handover motions, there are 752 (62.9%) samples where the conic model fits best, 442 (37.0%) where Decoupled Minimum Jerk model fits best, and 1 (0.08%) where Minimum Jerk trajectory fits best. When the conic model is the best fit, the average fitting error for the Decoupled Minimum Jerk model is 7.2 ± 10.0 times larger. However, when the Decoupled Minimum Jerk model is the best fit, the average fitting error for the Conic model is only 2.9 ± 3.6 times larger.

TABLE 5.1: Model fitting error mean and standard deviation.

	Conic	Decoupled Min Jerk	Min Jerk
Mean (mm)	2.0	2.4	14.3
Std. Deviation (mm)	3.2	3.9	12.4
Max (mm)	34.2	34.5	121.9

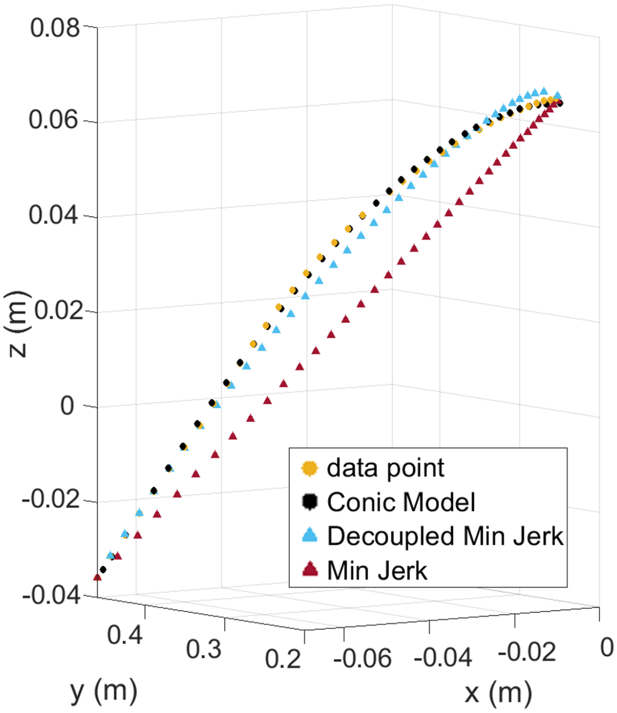


FIGURE 5.1: Example handover trajectory and the fitted models

The result from ANOVA comparing human handover reaching motion fitting error among the three models showed that there are significant differences ($F(2, 358) = 970.7, p < 0.0005$). Post hoc t-test revealed that both Decoupled Minimum Jerk and Conic model fitted unconstrained handover reaching motions better than Minimum Jerk trajectory ($t(1194) = 39.9, p < 0.0005; t(1194) = 33.7, p < 0.0005$). Furthermore, Conic model fitted better than Decoupled Minimum Jerk trajectory ($t(1194) = 3.03, p = 0.007$). Statistical analysis showed that the Conic model is the best fit model. Examining the fitted Conic model, 407 (34.1%) samples were elliptical, 705 (59.0%) hyperbolic, and 83 (6.9%) parabolic.

5.2 From Path to Trajectory

Path specifies the positional aspect of a motion, while trajectory specifies positional and temporal aspects of a motion. The Elliptical (Conic) model [29] is a path model while the Minimum Jerk [30] and Decoupled Minimum Jerk [28] models are trajectory

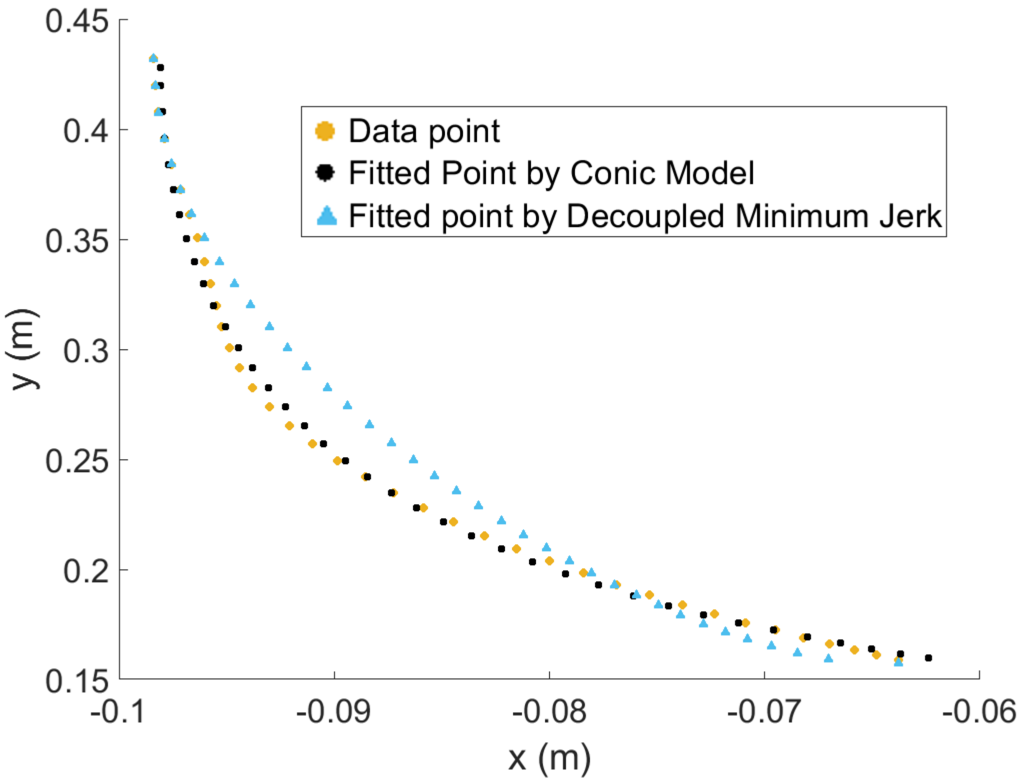


FIGURE 5.2: Fitted Conic and Decoupled Minimum Jerk models projected onto the best fit 2D plane from an example trial. Plots rotated to align conic major and minor axis with x and y axis.

models. Note that while θ in Eq. 4.2 can be expressed in terms of time to obtain a trajectory for the Conic model, previous study [29] had not examined the temporal aspect and considered only the path. This project also compared the path (without temporal aspect) of each motion model. Although the path generated by a Decoupled Minimum Jerk trajectory was found to fit the reaching motion well, we found that the error increases significantly if we use the actual trajectory and take the model's temporal aspect into consideration (i.e., defining point correspondence for Eq. 4.8 using time rather than closest point based on spatial distance). Hence, by finding an appropriate time parameterization of θ for Eq. 4.2, we can construct a Conic trajectory model that better fits human handover reaching motion as well. Furthermore, if a subtle pattern emerges in these fitted conic sections (their shapes and ranges of θ), we can exploit that to generate a fully defined trajectory only by a start point and an end point.

Chapter 6

Elliptical Trajectory Generator

Due to time limitation, although we mentioned that the Conic model could either be Hyperbola, Parabola or Ellipse, at the time this report is written, only the elliptical model has been used to formulate a trajectory generator. Thus, for analysis in this chapter, only samples in which the handover reaching trajectory is classified as ellipse are considered.

6.1 Hardware

This information is crucial to some of the design choices in later sections of this chapter. This project's ultimate goal is to formulate a trajectory generator for different models mentioned in earlier chapters to implement on a real-life robot, where the robot will perform a handover to a human receiver. The platform that we chose is a table-mounted Franka Emika Panda robotic arm. The robot has 7 degrees of freedom and a two-finger parallel gripper. The tabletop is $0.9m$ from the ground. Figure 6.1 shows the hardware setup we have in the laboratory.

6.2 Choosing Samples

Observing the dataset reveals that there are two θ patterns, depending on the participant's height and the object they were handing over. Figure 6.2 demonstrates the two

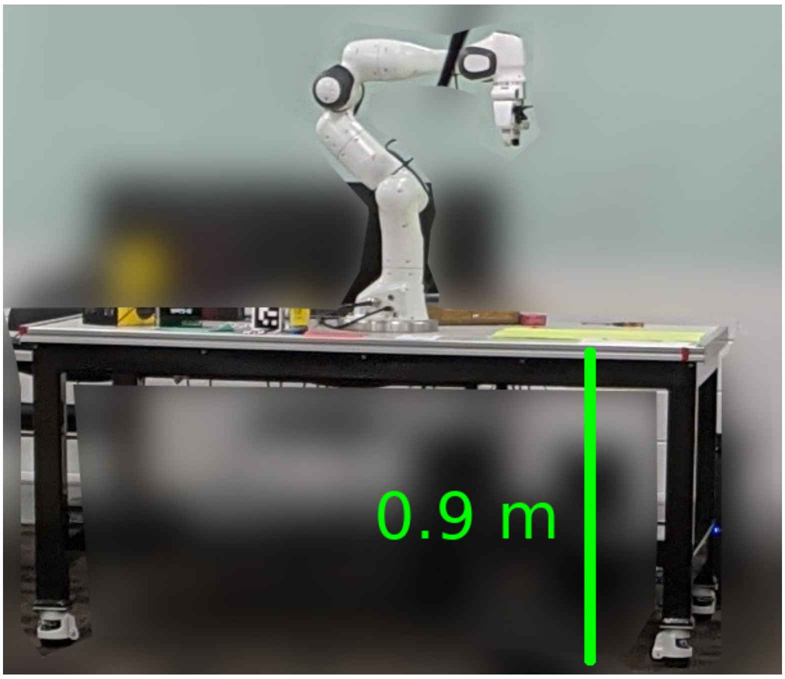
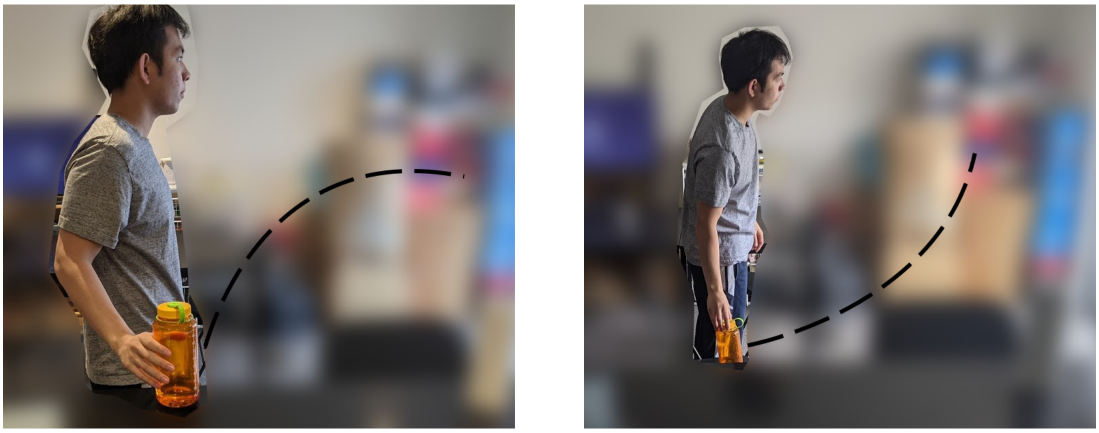


Figure 6.1: table-mounted Franka Emika Panda robotic arm

typical arcs. This is intuitive because this property of a curve is binary (either upper or lower). Within this dataset, human is more likely to adopt a lower arc trajectory the lower the position of the contact point between giver and the object and vice versa [1]. There may be an anatomical explanation for this phenomenon, but this is out of the project's scope. However, one can experiment with this behaviour by placing objects at different heights and perform a handover; there will be a certain level of awkwardness associated with one of the two arcs for a specific height.

For a given height, the object's position with respect to the human will also yield different motions. Figure 6.3 shows how a participant moves with different initial object placement. The dataset has three different initial placement for each object as mentioned in Section 3.1, left, right and behind. Because the givers are asked to perform unconstrained handover only with their right hand, getting objects on their left or behind would require twisting the body.

[1] The validity of this statement also depends on which approach the giver adopted between the two mentioned in Figure 3.5. This statement is only true if the approach in Figure 3.5a is adopted, which is 73.97% of the dataset. There was no correlation found if approach in Figure 3.5b is adopted



(A) Upper Arc

(B) Lower Arc

Figure 6.2: Two common arcs in handover. In the dataset, the table height is constant, and participants' heights vary. To simulate the same effect for visualization, objects are at different heights for the same person.

To reduce the complexity and identify a meaningful pattern to apply on a real-life robot, only samples where the object being initially placed on their right and the trajectory being upper arc are considered. Another reason for the latter choice is because of how the robot is set up in the laboratory: with the table height at 0.9m, people with average height tend to pick up the object with the same pose as Figure 6.2a and then execute an upper-arc trajectory.

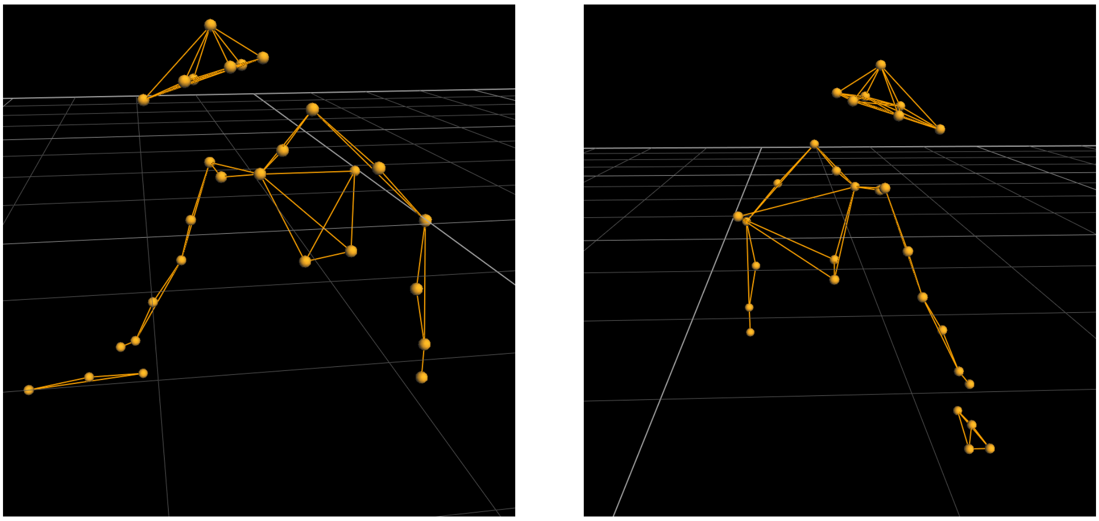
To summarize, within the scope of this report, the trajectory generator is formulated by analyzing samples such that:

- The handover reaching trajectory is classified as an ellipse
- The object is initially placed to the giver's right-hand side
- The trajectory follows an upper arc shape

These conditions yield 76 samples among 1195 samples from the dataset.

6.3 Trajectory Generator

The trajectory we are formulating is a reaching motion. This means that we do not account for the minute adjustment towards the end of the handover to place the object



(A) RHS

(B) LHS

Figure 6.3: Human motion with different initial object placement. A - object is initially placed on the Right Hand Side (RHS) of the giver, B - object is initially placed on the Left Hand Side (LHS) of the giver

on the human's hand correctly. Hence, the responsibility for a successful transfer relies on the human receiver or a perception system to track the human's hand and perform visual servoing once the trajectory is completed. This also means that this trajectory is only generated once the robot is ready to perform the handover, which is not necessarily at the instance the robot grasps the object. For the sake of simplicity, the starting point (handover-ready point) and the ending point (object transfer point) are the inputs to this trajectory generator. This assumption can be justified because we can obtain the object transfer point using an estimator [42], or simply the middle point of the receiver's torso, and the starting point depends on the object we are handing over and the configuration of the robot.

To fully define an ellipse in 2D, we have to calculate 5 parameters a, b, x_c, y_c, τ as described in Equation 4.2. With the inputs to the trajectory generator, we obtained 4 equations

$$\begin{cases} x_{start} = x_c + a\cos(\theta_{start})\cos(\tau) - b\sin(\theta_{start})\sin(\tau) \\ y_{start} = y_c + a\cos(\theta_{start})\sin(\tau) + b\sin(\theta_{start})\cos(\tau) \\ x_{end} = x_c + a\cos(\theta_{end})\cos(\tau) - b\sin(\theta_{end})\sin(\tau) \\ y_{end} = y_c + a\cos(\theta_{end})\sin(\tau) + b\sin(\theta_{end})\cos(\tau) \end{cases} \tag{6.1}$$

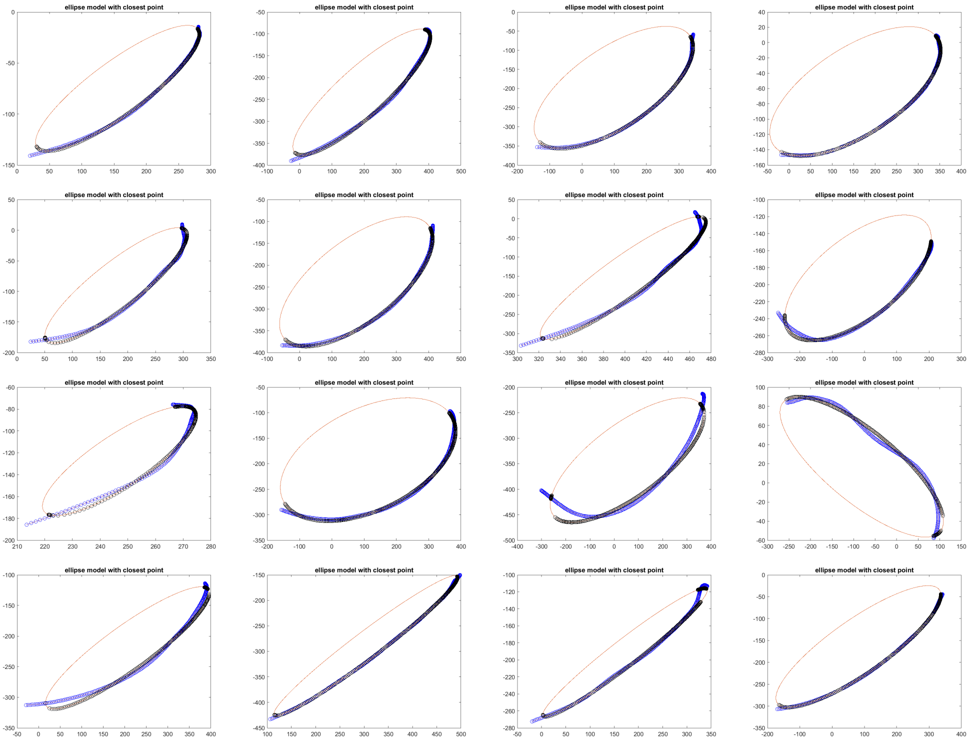


FIGURE 6.4: Random 16 ellipse fitting

In fact we have two more unknown: θ_{start}, and θ_{end}. In summary, we have four constraints and seven unknowns. This assumes that we already know the plane containing the ellipse.

6.3.1 Pattern Identification

The formulation so far has given us an underdetermined system, i.e. there are infinite solutions. However, by inspecting the dataset with the 124 samples chosen in section 6.2, we discover a few common patterns. Figure 6.4 shows 16 random samples of a fitted ellipse to handover reaching trajectory. From this figure, a pattern in θ_{start} and θ_{end} is quite visible. Upon further analysis, statistical results show that $\theta_{start} = 3.521° \pm 5.343°$ and $\theta_{end} = 179.023° \pm 4.687°$. We also noticed a pattern in the shape of the fitted ellipses, especially on its "thinness", i.e. the ratio of the semi-major and semi-minor of the ellipse. Statistical analysis shows that $\frac{a}{b} = 3.8399 \pm 2.6525$.

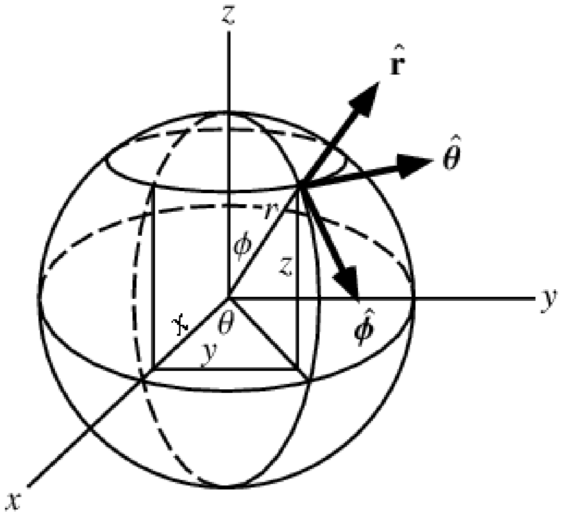


Figure 6.5: Spherical Coordinate

If the mean values of these three parameters are considered a solution, then the solution to the system of equations 6.1 is fully defined for a given starting point and ending point. However, this ellipse is only defined in a given plane, i.e. there are an infinite number of ellipses that satisfy Equations 6.1 in 3D space.

A plane in 3D is fully defined by one point on the plane and its normal vector, $\vec{n}$. Since we already know 2 (boundary) points on the plane, i.e. starting point and ending point. We only need to learn the pattern of the normal vector to identify the pattern of fitted planes. If we call the vector connecting the two boundary points boundary vector, $\vec{b}$, then the normal vector is always orthogonal to this boundary vector. In other words, the normal vector ($\vec{n}$) of the fitted plane ($\mathcal{P}$) lies on a plane, $\mathcal{Q}$, that is defined by the boundary vector, i.e. $\vec{b}$ is the normal vector of $\mathcal{Q}$. Hence, to find a pattern of the normal vector of $\mathcal{P}$, we have to express $\vec{n}$ in $\mathcal{Q}$. This constraint is crucial for reconstructing the plane's normal vector from an arbitrary boundary vector.

Fortunately, we do not have to manually define a set of bases on $\mathcal{Q}$ because there is already an established coordinate that does exactly this: Spherical Coordinate. Figure 6.5 gives an visualization of Spherical Coordinate. $\hat{\mathbf{r}}$ is unit vector for $\vec{b}$, $\hat{\theta}$ and $\hat{\phi}$ will span $\mathcal{Q}$. Figure 6.6 shows the result of the projections and the mean normal vector.

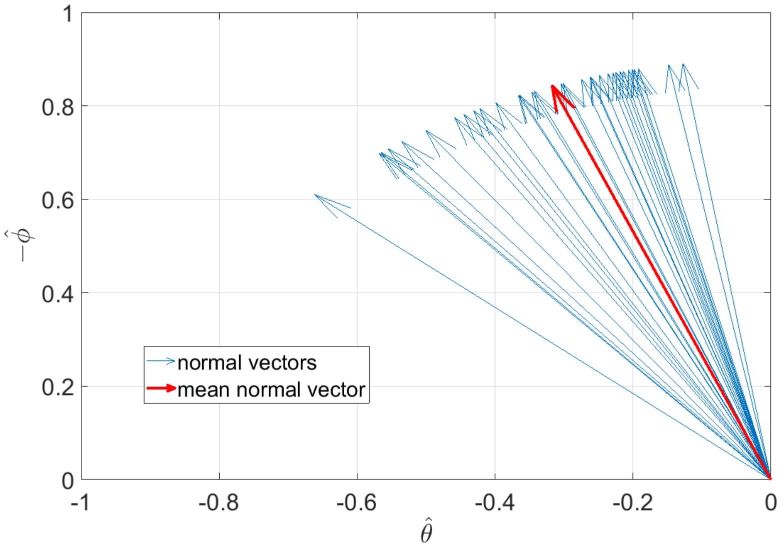


Figure 6.6: Fitted Normal Vector

6.3.2 Time parameterization

We have now obtained a well-defined path in 3D space for the elliptical model for a given start point and end point. To turn this into a trajectory, we have to introduce the temporal aspect to this generator. Since the elliptical model is parameterized by θ, we have to express θ as a function of time. Because we will ultimately implement this trajectory on the Franka robot, we must be careful about how to express this function. One might propose using the $\mathbf{\Theta}$ found in Algorithm 2, which is a function of time and fit a model to it. However, this will not be suitable for the robot. Since this dataset is unconstrained handover, multiple motions are blended, meaning the segmented handover has non-zero initial velocity and acceleration as discussed in Section 3.2. If we try to fit a model to $\mathbf{\Theta}$, it will result in a motion where initial velocity and acceleration is non-zero. This is a highly undesired behaviour for the robot as it will create a sudden jerk at the start of the handover, which potentially induce a high distrust and vigilance to the human receiver. Instead, with the collaboration of Sara Sheikholeslami, we obtained the dataset she collected in [29]. This dataset has already annotated the reaching motion, and in her study, the reaching motion has zero initial condition. Hence, we will use her dataset to fit a function of time to θ.

We normalize time so that all reaching motion starts from time 0 and end at 1 because

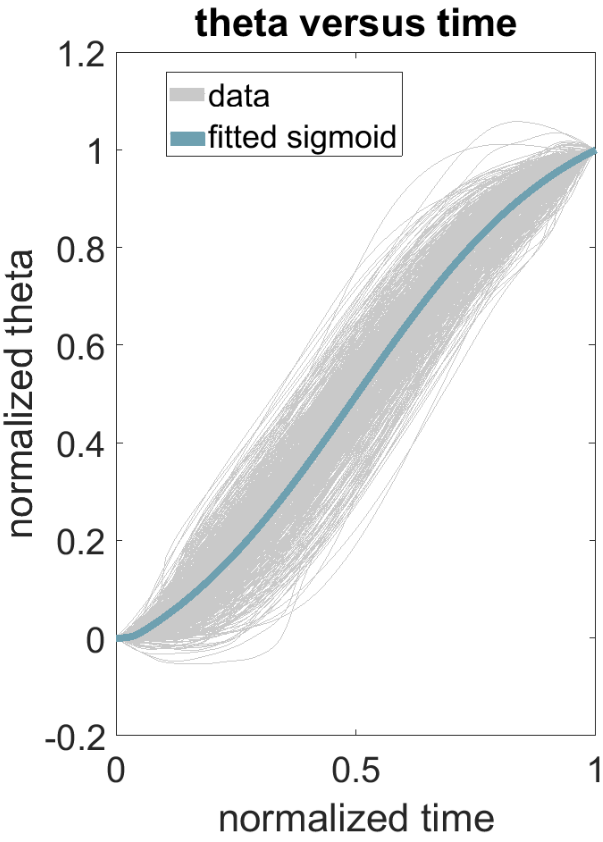


Figure 6.7: Fitted Theta

different reaching motions have different durations. We also normalize all θ so that they start from 0 and end at 1 because the range of θ depends on the length of the participant's arm. With this normalization, we notice that data have a high resemblance to a sigmoid function. Hence we tried to fit a sigmoid function to this normalized θ versus time dataset. A sigmoid in general form is expressed as

$$f(t) = \frac{l}{1 + e^{-m*(n-j)}} - o \tag{6.2}$$

To fit the sigmoid function, we first calculate the mean of θ versus time in Sara's dataset, interpolate the mean using `interp1` from MATLAB to N number of points, and use `fminsearch` where the input is the four parameters l, m, n, o. The cost function is the least square error similar to 4.23. Figure 6.7 shows the fitting result.

Chapter 7

User Study

7.1 User Study Design

A user study is designed to evaluate the performance of the Elliptical Motion model on natural human-robot handover. The study has been sent to and approved by Monash University's Ethics Review Board.

We will manipulate one independent variable, Trajectory Model: Elliptical Model, Decoupled Minimum Jerk Trajectory, slanted Decoupled Minimum jerk Trajectory [1] and Minimum Jerk Trajectory.

We will recruit participants within the lab and a limited number of people who have access to our building because of the COVID-19 restriction.

7.1.1 Procedure

Two investigators will supervise the user study in the university's laboratory. First, users will be asked to read the explanatory statement and sign a consent form. Next, we will explain the experiment by reading from a written script. The participants will then complete a small demographic survey.

[1] Decoupled Minimum Jerk Trajectory but rotated so that it lies on the same plane as the plane generated by the Elliptical model

Human-Robot Fluency
The human-robot team worked fluently together.
The robot contributed to the fluency of the interaction
Trust in Robot
I trusted the robot to do the right thing at the right time.
The robot was trustworthy
I felt safe completing the handovers
Working Alliance
The robot accurately perceives what my goals are.
I understand what the robot's goals are
The robot and I were working towards mutually agreed upon goals

Table 7.1: Survey Questions

Three objects, a steel water bottle, a multivitamin container, and a SPAM box, will be put at a designated place where the robot knows the location beforehand. After the robot picks up the object using Intel RealSenseD435 RGB-D camera that is mounted to the end-effector by detecting an AR tag attached to each object, the robot will move to ready pose to start the handover [2]. The user will handover 12 times in each trial: 4 times for each object. After each trial, the user will complete a survey. The survey questions are listed in Table 7.1 There are 3 trials per participant corresponding to the 3 Trajectory models, totalling to 36 handovers per participant. Finally, the participant will complete a post-study survey to provide comments and feedback on the study. An entire experiment takes approximately 30 minutes for each participant.

We utilize the subjective metrics outlined in [43]. We will only use a subset of questions that are relevant. All questions are measured on a 5 point Likert scale.

7.2 Robot Implementation

Because of hardware limitation, although the duration of a typical human handover is ≈ 1.2 seconds, the duration the robot execute each trajectory will be 4 seconds. When the user is ready for the first handover, a Kinect, which has already been calibrated the robot's base frame, will detect the skeletal pose of the participant and obtain the height

[2] The ready pose is different for each object because we pick the object top-down, so x and y position of the ready pose is constant while z will vary depending on object's height

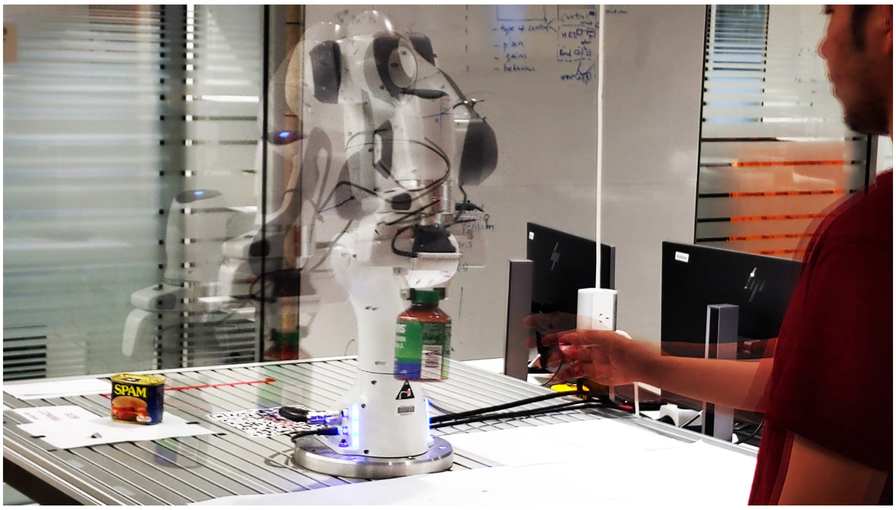


Figure 7.1: Elliptical Trajectory on Panda Franka Emika robotic arm

of their torso and the arm's length. We use these two pieces of information to estimate the Object Transfer point by setting the OTP' height to the participant's torso height, the reaching distance (from the base) by the length of the participant's arm. Hence, we obtain the starting point and ending point for each trajectory for each participant. This information is enough to generate all the 4 trajectories we want to compare as discussed in Section 6, Section 4.1.3, and Section 4.1.2.

Once the desired trajectory is loaded, the robot's end-effector will track it using a Jacobian-transpose Cartesian-impedance controller. Since we have not taken into account the object's orientation during handover, the end effector will have constant orientation throughout the reaching motion, which is $(0, 0, 0, 1)$ in quaternion, as shown in Figure 7.1.

Chapter 8

Discussion

8.1 Insights on differnce between models

The original Minimum Jerk trajectory model was formulated for reaching motions and follows a straight line trajectory for point-to-point motions. This was found to be contrary to natural human handover reaching motions, which were observed to be curved. The Decoupled Minimum Jerk model was subsequently proposed to allow a curved path, as inspecting the velocity profile along different axes of human reaching motions, it was found that the z-axis motion tends to terminate sooner than that in the xy plane, prompting the decoupling of the axes and setting different motion durations [28]. Consequently, the relationship between the motions along each axis is non-linear, creating a curve in space instead of a straight line. Although this approach is more similar to natural human motions, the residual at the end of the Decoupled Minimum Jerk trajectory yields an unnaturally straight line at the end of the motion. For example, without loss of generality, let $t_z < t_{xy}$. At t_z, the motion along the z-axis has finished while motion in the xy plane has not. After this, motion along the z-axis simply ceases, while the reaching motion model continues to move in the xy plane in a straight line. Fig. 8.1 shows the effects of t_z to t_{xy} ratio on the shape of the curve. The Conic model, on the other hand, does not suffer from this phenomenon. Hence, of the three models considered, the Conic model is the only one that yields curved motion along the entire reaching trajectory.

We have chosen the Minimum Jerk Model as the baseline since it is a well-established model for handover trajectories. Although the conic model's better fit may be attributed

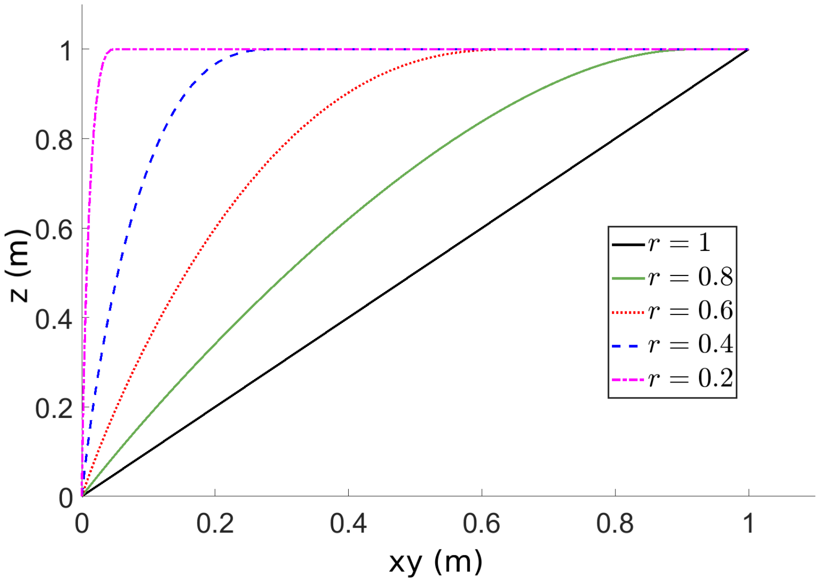


Figure 8.1: Resulting Decoupled Minimum Jerk Trajectories for different duration ratios $r = \frac{t_z}{t_xy}$. Note that $r = 1$ yields a Minimum Jerk Trajectory

to having more parameters, and one may be tempted to simply use an even higher-order function to achieve a better fit, this would increase the computational cost for online trajectory generation and risk overfitting. We have also confirmed by Bayesian Evidence (following [29, 44]) that indeed, a second-order polynomial (conic) best fits the dataset.

8.2 Unconstrained Handovers

Unlike most existing studies on reaching motions, which focused on constrained (seated, tabletop) tasks, we have examined unconstrained handovers, as we are interested in enabling handovers in a more general setting. This meant that participants were no longer restricted to using only arm motion, but could utilize their whole body. We discovered that, contrary to literature reporting that human reaching motion in constrained tasks is elliptical [29], a majority of human reaching motion in unconstrained handovers are hyperbolic instead. We observed that unconstrained participants naturally utilize full-body motion to carry out handovers, with a combination of the following body motions used (in addition to arm motion).

Torso Twisting: Objects were placed beside or behind the giver. As a result, the giver naturally needs to turn their body to pick up the objects. However, torso twisting was observed to occur not only during the pick-up phase, but throughout the entire handover task. Participants used torso twisting to bring the object from the side/back to the front (towards the receiver) and extend their arm's reach when delivering the object to the receiver. Hence, the hand trajectory incorporates torso rotational motion and shoulder translational motion.

Leaning: Towards the end of the handover, the giver sometimes leans towards the receiver to deliver the object. This leaning motion can potentially be pre-planned, or if the giver misjudged the object transfer location after the arm has been fully reached out and torso twisting utilized, the giver needs to lean further to cover the remaining gap.

Stepping: Givers occasionally take steps towards the receiver during handover. This potentially adds a translational component to the hand trajectory with respect to the world frame. Consequently, the end of the hand trajectory may be elongate and flatten. However, literature shows that givers tend to begin their reach only after they are sufficiently close to the receiver [11], and in our analysis, we found that most givers took zero or only one step. Hence, the observed trajectories in our studies are mostly a combined result of the other aforementioned motions.

8.3 Rare Handover Cases

We observed some unexpected/outlier behaviour in the dataset. For example, some givers only picked up the object and held the object near their torso, without much "reaching" motion towards the receiver, and waited for the receiver to reach out to take the object. Another example is that sometimes the giver has to "hand down" the object because the receiver did not reach out their hand sufficiently to meet the giver at the midpoint between them. These interesting outliers may provide insights on human behaviour and how we should program human-robot handovers for humanoids operating in the real world.

8.4 Towards Fluent Human-Robot Handovers

Service robots will be expected to perform frequent handovers with users in many applications. Hence, their ability to perform this task well is crucial. Existing studies tend to argue the use of a Minimum Jerk or Decoupled Minimum Jerk trajectory for handover reaching motions by showing that their velocity profiles look similar to observed human reaching motions [15, 16, 28]. Furthermore, they show that when implemented onto robot handover reaching motions, it produces a more positive subjective perception of the robot. However, they have not fitted the proposed trajectory models to human reaching motions. We have empirically shown that the Conic model fits human handover reaching motions better than the existing Minimum Jerk and Decoupled Minimum Jerk trajectories.

Furthermore, existing studies have been primarily restricted to constrained (seated, tabletop) tasks. However, service robots will need to perform handovers in a wide range of unconstrained situations. We have shown that human reaching motions in unconstrained handovers, while still conical, is no longer restricted to elliptical, as in constrained tasks [29]. There is also coordination of full-body motion.

As we observed that humans use coordinated full-body motion for handovers, a robot giver may also need to consider how to coordinate its full-body motion, instead of planning only arm motion, as most existing handover motion planners do [15, 16, 28]. As a receiver, the robot can observe the onset of the human giver's reach and use the Conic model to estimate the endpoint to predict the point of object transfer. This will then allow greater flexibility and enable the robot to plan and begin executing its reaching motion for intercepting the object early on during the reaching phase of the handover, unlike existing robots that tend to wait until the giver finishes reaching, or can only hand over the object at a predefined location [18, 45, 46]. As handover is a ubiquitous task for service robots working with people, improving their competence in this fundamental task is expected to improve their overall interaction with users in a wide range of applications.

8.5 Limitation and Future Work

As mentioned in this footnote 2 on page 12, if the conic section is classified as a parabola in this project, it is fitted with a hyperbola due to numerical zero. While this is still valid, it is not consistent with the method we proposed. Therefore, further research into the literature is required to find the parametric form of a parabola from a general conic section's coefficients.

Among the three possible Conic Section: ellipse,s hyperbola, and parabola (we did not find a sample where circle fits best), we have only formulated a trajectory generator for ellipses. Further analysis is required to formulate the rest.

We have only looked at a very specific condition for the elliptical trajectory generator: a right-hand upper-arc trajectory. We can simply mirror the trajectory and simulate a motion that hand objects initially placed on the left-hand side by a "left-handed" robot. However, further analysis is required to find a pattern where the robot is "right-handed", and the object is initially placed on the left.

For the current hardware setup, the robot is fundamentally anatomically different from a human arm. The robot is elbow-up while the human is elbow-down. If we can set up the robot so that it has the configuration in [47], the motion would look more human-like.

The statistical analysis to formulate the trajectory generator is not rigorous enough (how we determine the θ_{start}, θ_{end}, $\frac{a}{b}$, mean $\vec{n}$). Further analysis would require to confirm if the pattern has significant results. a and b might have other relationship than just simply proportional.

Since we found a correlation between the object's placement and the tendency to adopt upper-arc or lower-arc trajectory, further analysis can reveal a concrete relationship between these two characteristics.

The user study is already designed and approved by the Monash Board of Ethics. So naturally, future work would include carrying out the user study.

Chapter 9

Conclusion

This project has experimentally compared the existing Minimum Jerk, Decoupled Minimum Jerk, and Conic models for unconstrained human handover reaching motions. We showed that while the Conic models fit best, unlike seated tabletop reaching motions that were elliptical primarily, there is a split between elliptical and hyperbolic motion models. This project provided experimental validation of reaching models for human handover reaching motions. Results suggest that the Conic model may be used to generate more humanlike motions compared to the well-established Minimum Jerk models. Furthermore, unlike solo reaching motions which are mostly elliptical, a mix of elliptical and hyperbolic motions should be expected in handovers.

We have also designed a user study to compare the performance of Elliptical Models to Minimum Jerk, Decoupled Minimum Jerk, and Minimum Jerk. A trajectory generator for the Elliptical model was also formulated with a specific condition relevant to our user study. We have also implemented and tested all these trajectories on the Franka Emika Panda robotic arm.

There are a lot of potential directions to improve and expand from what we discovered from this project. And they are all fascinating.

Appendix A

(Decoupled) Minimum Jerk Coefficients

The equation for minimum jerk trajectory is typically defined as a function of time in 1D, $r(t)$, as described in equation 4.5. Since $r(t)$ is a polynomial, its first and second order derivative is straightforward:

$$\begin{aligned} r'(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ r''(t) &= 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{aligned}$$

To describe this trajectory in 3D space with arbitrary start poind and end point, we can express $r(t)$ in three bases of the coordinate system:

$$\mathbf{r}(t) = (x(t), y(t), z(t))$$

where

$$\begin{aligned} x(t) &= a_{0x} + a_{1x} t + a_{2x} t^2 + a_{3x} t^3 + a_{4x} t^4 + a_{5x} t^5 \\ y(t) &= a_{0y} + a_{1y} t + a_{2y} t^2 + a_{3y} t^3 + a_{4y} t^4 + a_{5y} t^5 \\ z(t) &= a_{0z} + a_{1z} t + a_{2z} t^2 + a_{3z} t^3 + a_{4z} t^4 + a_{5z} t^5 \end{aligned}$$

Without loss of generality, instead of finding coefficients for each dimension, we can establish the equations for coefficients of $r(t)$. In the context of handover, and as we observed from the dataset, terminal velocity and acceleration is 0, but initial velocity and acceleration can be non-zero.

Hence, we have:

$$\begin{aligned}
r(0) = r_0 &= a_0 + a_1(0) + a_2(0)^2 + a_3(0)^3 + a_4(0)^4 + a_5(0)^5 = a_0 \\
r'(0) = v_0 &= a_1 + 2a_2(0) + 3a_3(0)^2 + 4a_4(0)^3 + 5a_5(0)^4 = a_1 \\
r''(0) = a_0 &= 2a_2 + 6a_3(0) + 12a_4(0)^2 + 20a_5(0)^3 = 2a_2 \\
r(t_e) = r_e &= a_0 + a_1 t_e + a_2 t_e^2 + a_3 t_e^3 + a_4 t_e^4 + a_5 t_e^5 \\
r'(t_e) = v_e &= 0 = a_1 + 2a_2 t_e + 3a_3 t_e^2 + 4a_4 t_e^3 + 5a_5 t_e^4 \\
r''(t_e) = a_e &= 0 = 2a_2 + 6a_3 t_e + 12a_4 t_e^2 + 20a_5 t_e^3
\end{aligned}$$

where t_e is the duration, r_0 is the initial position, r_e is the final position, v_0 is the initial velocity, v_e is the final velocity, a_0 is the initial acceleration, and a_e is the final acceleration. a_3, a_4, and a_5 is now 3 unknowns with a system of 3 linear equations. Hence, the solution for the coefficients can be found as

$$\begin{aligned}
a_0 &= r_0 \\
a_1 &= v_0 \\
a_2 &= \frac{a_0}{2} \\
a_3 &= -\frac{3}{2}\frac{a_0}{t_e} - 6\frac{v_0}{t_e^2} + 10\frac{(r_e - r_0)}{t_e^3} \\
a_4 &= \frac{3}{2}\frac{a_0}{t_e^2} + 8\frac{v_0}{t_e^3} + -15\frac{(r_e - r_0)}{t_e^4} \\
a_5 &= -\frac{1}{2}\frac{a_0}{t_e^3} - 3\frac{v_0}{t_e^4} + +6\frac{(r_e - r_0)}{t_e^5}
\end{aligned}$$

It is transparent that the coeffcients depdends on the boundary condition and the duration of the trajectory. Notice that if $t_{e_x} = t_{e_y} = t_{e_z}$, we have minimum jerk trajectory. If $t_{e_x} = t_{e_y} \neq t_{e_z}$, we have decoupled minimum jerk trajectory as described in section 4.1.3.

Appendix B

Rodrigues' Rotation Formula

To align $\hat{n}$ and $\hat{k}$, we have to choose a rotation axis and the angle to rotate. Since $\hat{n}$ and $\hat{k}$ is already unit vector, the angle between $\hat{n}$ and $\hat{k}$, α, can be found with simple calculus

$$\alpha = cos^{-1}(\hat{n} \cdot \hat{k})$$

The rotation axis, $\hat{r}$, to rotate by α to align $\hat{n}$ and $\hat{k}$, is a vector such that it is orthogonal to both $\hat{n}$ and $\hat{k}$

$$\hat{r} = \hat{n} \times \hat{k} = (r_1, r_2, r_3)$$

We can then construct the skew-symmetric matrix corresponding to $\hat{r}$ as

$$[\hat{r}] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$$

Rotation matrix $\mathbf{R}$ can then be constructed as

$$\mathbf{R} = \mathbf{I} + \sin(\alpha)[\hat{r}] + (1 - \cos(\alpha))[\hat{r}]^2$$

References

[1] A. Badii, I. Etxeberria, C. Huijnen, M. Maseda, S. Dittenberger, A. Stainer-Hochgatterer, D. Thiemert, and A.-S. Rigaud, "Companionable: Graceful integration of mobile robot companion with a smart home environment," *Gerontechnology*, vol. 8, pp. 181–181, 07 2009.

[2] W. K. Fung, Y. Y. Leung, M. K. Chow, Y. H. Liu, Y. Xu, W. Chan, T. W. Law, S. K. Tso, and C. Y. Wang, "Development of a hospital service robot for transporting task," in *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003*, vol. 1, 2003, pp. 628–633 vol.1.

[3] W. P. Chan, G. Hanks, M. Sakr, T. Zuo, H. F. M. Van der Loos, and E. Croft, "An augmented reality human-robot physical collaboration interface design for shared, large-scale, labour-intensive manufacturing tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020. Proceedings. 2020*, 2020.

[4] H. Masuzawa, J. Miura, and S. Oishi, "Development of a mobile robot for harvest support in greenhouse horticulture—person following and mapping," in *2017 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2017, pp. 541–546.

[5] R. Wilcox, S. Nikolaidis, and J. Shah, "Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing," in *Robotics: Science and Systems*, 2012.

[6] Y. S. Choi, T. Chen, A. Jain, C. Anderson, J. D. Glass, and C. C. Kemp, "Hand it over or set it down: A user study of object delivery with an assistive mobile manipulator," in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, Sep 2009, p. 736–743.

[7] C. Shi, M. Shiomi, C. Smith, T. Kanda, and H. Ishiguro, “A model of distributional handing interaction for a mobile robot,” in *Robotics: Science and Systems IX.* Robotics: Science and Systems Foundation, Jun 2013. [Online]. Available: http://www.roboticsproceedings.org/rss09/p55.pdf

[8] V. Ortenzi, A. Cosgun, T. Pardi, W. Chan, E. Croft, and D. Kulic, “Object handovers: a review for robotics,” *arXiv:2007.12952 [cs, eess]*, Jul 2020, arXiv: 2007.12952. [Online]. Available: http://arxiv.org/abs/2007.12952

[9] W. P. Chan, C. A. Parker, H. M. Van Der Loos, and E. A. Croft, “A human-inspired object handover controller,” *International Journal of Robotics Research*, vol. 32, no. 8, p. 971–983, 2013.

[10] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard, “A human-inspired controller for fluid human-robot handovers,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, p. 324–331.

[11] P. Basili, M. Huber, T. Brandt, S. Hirche, and S. Glasauer, *Investigating Human-Human Approach and Hand-Over*, ser. Cognitive Systems Monographs. Springer, 2009, p. 151–160. [Online]. Available: https://doi.org/10.1007/978-3-642-10403-9_16

[12] A. Moon, D. M. Troniak, B. Gleeson, M. K. Pan, M. Zeng, B. A. Blumer, K. MacLean, and E. A. Croft, “Meet me where i’m gazing: how shared attention gaze affects human-robot handover timing,” in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction - HRI ’14.* ACM Press, Mar 2014, p. 334–341.

[13] W. P. Chan, C. A. C. Parker, H. F. M. Van der Loos, and E. A. Croft, “Grip forces and load forces in handovers: Implications for designing human-robot handover controllers,” in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar 2012, p. 9–16.

[14] K. Yamane, M. Revfi, and T. Asfour, “Synthesizing object receiving motions of humanoid robots with human motion database,” in *International Conference on Robotics and Automation.* Ieee, May 2013, p. 1629–1636. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6630788

[15] S. Shibata, B. M. Sahbi, K. Tanaka, and A. Shimizu, "An analysis of the process of handing over an object and its application to robot motions," in *International Conference on Systems, Man, and Cybernetics*, Apr 1997, p. 64–69.

[16] M. Huber, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer, "Human-robot interaction in handing-over tasks," in *International Symposium on Robot and Human Interactive Communication*, Aug 2008, p. 107–112.

[17] C. Becchio, L. Sartori, and U. Castiello, "Toward you: The social side of actions," *Current Directions in Psychological Science*, vol. 19, no. 3, p. 183–188, Jun 2010.

[18] A. Kshirsagar, M. Lim, S. Christian, and G. Hoffman, "Robot gaze behaviors in human-to-robot handovers," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, p. 6552–6558, Oct 2020.

[19] A. H. Mason and C. L. MacKenzie, "Grip forces when passing an object to a partner," *Experimental Brain Research*, vol. 163, no. 2, p. 173–87, May 2005.

[20] S. Parastegari, E. Noohi, B. Abbasi, and M. Žefran, "A fail-safe object handover controller," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, p. 2003–2008.

[21] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S. S. Srinivasa, M. Cakmak, and V. Micelli, "Towards seamless human-robot handovers," *Journal of Human-Robot Interaction*, vol. 1, no. 1, p. 1–23, 2013.

[22] M. K. Pan, V. Skjervøy, W. P. Chan, M. Inaba, and E. A. Croft, "Automated detection of handovers using kinematic features," *The International Journal of Robotics Research*, vol. 36, no. 5–7, p. 721–738, Jun 2017.

[23] J. Aleotti, V. Micelli, and S. Caselli, "An affordance sensitive system for robot to human object handover," *International Journal of Social Robotics*, vol. 6, no. 4, p. 653–666, Jun 2014.

[24] M. Cakmak, S. S. Srinivasa, M. K. Lee, S. Kiesler, and J. Forlizzi, "Using spatial and temporal contrast for fluent robot-human hand-overs," in *International Conference on Human Robot Interaction*, 2011, p. 489–496.

[25] C. G. Atkeson and J. M. Hollerbach, "Kinematic features of unrestrained vertical arm movements," *Journal of Neuroscience*, vol. 5, no. 9, p. 2318–2330, Sep 1985.

[26] P. Morasso, “Spatial control of arm movements,” *Experimental Brain Research*, vol. 42, no. 2, p. 223–227, Apr 1981.

[27] W. Abend, E. Bizzi, and P. Morasso, “Human arm trajectory formation,” *Brain*, vol. 105, no. 2, p. 331–348, Jun 1982.

[28] M. Huber, H. Radrich, C. Wendt, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer, “Evaluation of a novel biologically inspired trajectory generator in human-robot interaction,” in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, Sep 2009, p. 639–644.

[29] S. Sheikholeslami, G. Lee, J. W. Hart, S. Srinivasa, and E. A. Croft, *A Study of Reaching Motions for Collaborative Human-Robot Interaction*, ser. Springer Proceedings in Advanced Robotics. Springer International Publishing, 2020, vol. 11, p. 584–594. [Online]. Available: http://link.springer.com/10.1007/978-3-030-33950-0_50

[30] N. Hogan, “Control and coordination of voluntary arm movements,” in *1982 American Control Conference*, Jun 1982, p. 522–528.

[31] T. Flash and N. Hogan, “The coordination of arm movements: an experimentally confirmed mathematical model,” *Journal of Neuroscience*, vol. 5, no. 7, p. 1688–1703, Jul 1985.

[32] Z. Li and K. Hauser, “Predicting object transfer position and timing in human-robot handover tasks,” *Science and Systems*, p. 38, 2015.

[33] C. T. Landi, Y. Cheng, F. Ferraguti, M. Bonfè, C. Secchi, and M. Tomizuka, “Prediction of human arm target for robot reaching movements,” in *IROS*, 2019, pp. 5950–5957.

[34] R. Marteniuk, C. Mackenzie, M. Jeannerod, S. Athenes, and C. Dugas, “Constraints on human arm movement trajectories.” *Canadian Journal of Psychology/Revue canadienne de psychologie*, vol. 41, no. 3, p. 365–378, 1987.

[35] W. P. Chan, M. K. X. J. Pan, E. A. Croft, and M. Inaba, “An affordance and distance minimization based method for computing object orientations for robot human handovers,” *International Journal of Social Robotics*, vol. 12, no. 1, p. 143–162, Jan 2020.

[36] J. R. Fanchi, *Math refresher for scientists and engineers*, 3rd ed. Hoboken, NJ: Wiley, 2006, p. 365–378.

[37] C. F. Van Loan, "Using the ellipse to fit and enclose data points," *Department of Computer Science Cornell University*, p. 54, 2008.

[38] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 37–52, 1987, proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0169743987800849

[39] G. Williams, "Overdetermined systems of linear equations," *The American Mathematical Monthly*, vol. 97, no. 6, pp. 511–513, 1990.

[40] A. B. Ayoub, "The central conic sections revisited," *Mathematics Magazine*, vol. 66, no. 5, pp. 322–325, 1993. [Online]. Available: http://www.jstor.org/stable/2690513

[41] ——, "The eccentricity of a conic section," *The College Mathematics Journal*, vol. 34, no. 2, pp. 116–121, 2003. [Online]. Available: http://www.jstor.org/stable/3595784

[42] H. Nemlekar, D. Dutia, and Z. Li, "Object transfer point estimation for fluent human-robot handovers," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2627–2633.

[43] V. Ortenzi, A. Cosgun, T. Pardi, W. Chan, E. Croft, and D. Kulic, "Object handovers: A review for robotics," *arXiv preprint arXiv:2007.12952*, 2020.

[44] T. Minka, "Bayesian linear regression," Citeseer, Tech. Rep., 2000.

[45] V. Micelli, K. Strabala, and S. Srinivasa, "Perception and control challenges for effective human-robot," in *Robotics: Science and Systems*, 2011. [Online]. Available: http://repository.cmu.edu/robotics/1026/?utm_source=repository.cmu.edu%2Frobotics%2F1026&utm_medium=PDF&utm_campaign=PDFCoverPages

[46] W. P. Chan, K. Nagahama, H. Yaguchi, Y. Kakiuchi, K. Okada, and M. Inaba, "Implementation of a framework for learning handover grasp configurations through observation during human-robot object handovers," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, p. 1115–1120.

[47] M. K. Pan, E. Knoop, M. Bächer, and G. Niemeyer, "Fast handovers with a robot character: Small sensorimotor delays improve perceived qualities," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6735–6741.