Classification of Co-Manipulation Modus with Human-Human Teams for Future Application to Human-Robot Systems

SETH FREEMAN, SHADEN MOSS, JOHN L. SALMON, and MARC D. KILLPACK,
Brigham Young University, Provo, Utah, USA

Despite the existence of robots that can physically lift heavy loads, robots that can collaborate with people to move heavy objects are not readily available. This article makes progress toward effective human-robot co-manipulation by studying 30 human-human dyads that collaboratively manipulated an object weighing 27 kg without being co-located (i.e., participants were at either end of the extended object). Participants maneuvered around different obstacles with the object while exhibiting one of four modi–the manner or objective with which a team moves an object together–at any given time. Using force and motion signals to classify modus or behavior was the primary objective of this work. Our results showed that two of the originally proposed modi were very similar, such that one could effectively be removed while still spanning the space of common behaviors during our co-manipulation tasks. The three modi used in classification were *quickly*, *smoothly* and *avoiding obstacles*. Using a deep convolutional neural network (CNN), we classified three modi with up to 89% accuracy from a validation set. The capability to detect or classify modus during co-manipulation has the potential to greatly improve human-robot performance by helping to define appropriate robot behavior or controller parameters depending on the objective or modus of the team.

CCS Concepts: • **Human-centered computing** → **Collaborative interaction**; Empirical studies in HCI; *HCI theory, concepts and models*; • **Computing methodologies** → *Neural networks*; • **Computer systems organization** → *Robotic control*;

Additional Key Words and Phrases: Modus, physical human-robot interaction, co-manipulation

ACM Reference format:
Seth Freeman, Shaden Moss, John L. Salmon, and Marc D. Killpack. 2024. Classification of Co-Manipulation Modus with Human-Human Teams for Future Application to Human-Robot Systems. *ACM Trans. Hum.-Robot Interact.* 13, 4, Article 50 (October 2024), 26 pages.
https://doi.org/10.1145/3659059

1 Introduction

Robotics and automated systems have permanently altered many industries around the world by lifting heavy payloads, performing precision welding, doing pick and place maneuvers, or

We would like to thank the National Science Foundation's National Robotics Initiative for making this work possible. This work was funded under grant number 2024792. We would also like to thank Lesa Stern Ph.D., Mark Callister Ph.D. and Kevin John Ph.D. for their help in the experiment design.
Authors' Contact Information: Seth Freeman, Brigham Young University, Provo, Utah, USA; e-mail: sethmfree@gmail.com; Shaden Moss, Brigham Young University, Provo, Utah, USA; e-mail: engineering.moss@gmail.com; John L. Salmon, Brigham Young University, Provo, Utah, USA; e-mail: johnsalmon@byu.edu; Marc D. Killpack (Corresponding author), Brigham Young University, Provo, Utah, USA; e-mail: marc_killpack@byu.edu.

ACM 2573-9522/2024/10-ART50
https://doi.org/10.1145/3659059

accomplishing many other repetitive but important tasks. However, while these robots are useful, they are not intelligent enough to operate in unstructured environments in the real world. Robots generally require that their working environment be void of human disturbances or interaction. Efforts to make robots more fit for the real world have led to the development of compliant and soft robots that are safer for working with humans. However, while the hardware for real-world robot interaction continues to improve, a critical deficit in robot intelligence remains. This deficit is the intelligence necessary to enable robots to intuitively interact with humans, human teams, and human environments in natural and effective ways.

Specific to this article, we propose that the intelligence required for a robot to assist a human team in co-manipulating an object in a natural and comfortable way is currently underdeveloped, especially for objects that are longer than a few inches in length. Current co-manipulation control algorithms do not reach human levels of performance in detecting a partner's intended path or manner of motion. This type of intelligence will certainly be required for any sort of home or office assistant robot and will benefit many other fields like construction, medical evacuation, and search and rescue. The Bureau of Labor Statistics has reported that over 40% of workplace injuries result from overexertion and that overexertion from lifting was the most common cause [60]. Robots that can safely and intelligently carry objects with a human partner can make a big impact in the lives of these workers. The intent of this research is to make progress toward this greater intelligence by developing the ability to identify the intended manner in which a team co-manipulating an object wants to move. Our specific contributions are described in Section 1.3.

1.1 Problem Motivation

There is a scarcity of research that considers evaluating team performance for co-manipulation tasks based on objectives, priorities, or intent of the team (beyond merely completing the task). When a team performs an action, before it can be said whether they were successful in performing that action, their intended manner of performing the action must be known. In the realm of robotics and controls it is tempting to pick a canonical metric on which to judge performance, such as minimum energy trajectories or completion time, because they are well-known metrics. However, these metrics should only be applied in a principled manner. It would be best if the intended objective (both the goal state *and* the desired manner of reaching that goal) of the team is known before deciding which metric to use. Previous researchers have discovered that if task completion is the only objective given to a team co-manipulating an object, it is difficult to compare teams in a principled manner. Thus, it is difficult to ascertain which objective function should be used to evaluate the team's performance [35–37]. However, evaluating a human-human team's performance for a given task is essential to enabling robots to reproduce or mimic that behavior.

For instance, if a team's intended manner was to complete the task quickly, then minimum energy would be the wrong metric to use to judge the performance of the team. If another team's intended manner was to be very careful because the object they were moving was delicate or fragile, neither completion time nor minimum energy would be the right metric to consider for evaluating performance.

The improper implementation of a performance metric and intended manner of motion in an algorithm can be disastrous if a robot replaces one of the members of a team co-manipulating an object. For example, in a search-and-rescue situation, without concern for the intended manner of motion, a robot could move an object very smoothly and with minimum jerk, achieving "optimal performance," but completely fail to meet the team's desire of moving quickly. Alternatively, in another scenario, if a robot was optimized for speed, it might exert forces that could damage fragile payloads, or cause injury to human team members.

1.2 Problem Terminology

Another way to define the "intended manner" of the team is the word "modus," with the plural "modi." One definition of the word modus is "a mode of procedure, a way of doing something" [1]. In this article we will use modus to express the "intended manner" in which a team moves an object through an environment, more accurately describing their overall behavior. Examples of modi include: moving quickly, moving carefully because of a special consideration of the environment (e.g., fragile drywall, unstable rubble), moving carefully because the object itself is fragile or needs special moving considerations (e.g., a person with a spinal injury), and moving "normally" with no special care to the environment or the completion time. In this article, the word "modus" will be used to define high-level behavior and objectives; whereas the trajectory chosen, or the final configuration of the object, will not be considered part of the modus. Section 3.1 describes the subset of potential modi that we used in our experiments in more detail, including motivation and justification.

We will refer to the act of a team moving an object together as physical "co-manipulation" to stay consistent with previous research [35, 36]. Co-manipulation simply refers to a team of people collaboratively manipulating or moving an object together and involves not only the motion, but also the interaction forces and spoken and unspoken communication of the team. In particular, this research involves the co-manipulation of objects that are of a significant dimension and weight which would require more than one agent to successfully complete the task.

1.3 Research Questions and Contributions

The purpose of this work is to answer two important questions. First, when the modus of a team varies, will the behavior of the team change enough to be observable using common sensors on robotic platforms? Second, if the change in behavior with varying modi is observable with certain signals, can those observations be used to classify the modus of the team at any given time?

If the modus of a team can be classified from externally observable data, it could have a substantial impact on the field of **physical human-robot interaction (pHRI)**. It would enable the development of a principled manner of identifying which teams are the highest performers. For example, if it is known that a team was intending to move quickly there are metrics associated with the quickly modus (e.g., maximum velocity, average velocity, completion time, and so on) which could be used to evaluate the team's performance. If it is possible to determine if a single team has performed well, it is also possible to compare teams against each other and evaluate which teams perform best.

Furthermore, these high-performing teams can be studied and used as models to develop robotic partners. Future robot controllers could be designed to classify team behavior or modus, and use the classification to optimize the robot's performance to mimic the behavior of the best-performing human partners while also moving the object to a desired goal state. For example, if a robot has determined that the human is being careful with the object, the robot can alter its control method to optimize for slow and smooth motion. This allows the robot to quantify its own performance and improve its behavior online. Importantly, the method must be able to run in real time as the robot will need to constantly monitor the signals from their human partner to notice any possible shifts in modus. We theorize that humans can change their modus based on the object being co-manipulated, the obstacles they face, the team's fatigue, and many other reasons.

With the aforementioned research questions and high-level strategy described, the fundamental contributions of this article are:

(1) Development of a **neural network (NN)** architecture to classify the modus of a co-manipulation task in real time

(2) Identification of a subset of important modi that are both identifiable and relevant for co-manipulation
(3) Analysis of the most effective signals for modus classification during co-manipulation
(4) An open-source dataset obtained from the instrumentation, observation, and recording of a large number of dyadic teams manipulating an extended object through and around physical obstacles

2 Related Work

For decades people have theorized about how to enable robots to assist people in their real-world tasks. Around the end of the 20th century, a number of designs enabled robots to assist humans in co-manipulation tasks. These designs included one arm on a mobile base [27], a humanoid robot with two arms on a mobile base [29], and a humanoid robot with two arms and two legs [62]. Since then, the field of human-robot co-manipulation has only continued to develop. However, most of the developments have been focused on developing robot hardware to perform co-manipulation tasks. Our work concerns modi, which will enable the principled selection of metrics to aid in evaluating robot performance to determine how well the robots are performing a given task with a given context.

2.1 Modus

While the term "modus" might be new in the field of co-manipulation, there is a small body of research closely related to the topic. For example, there have been a series of studies performed by Shaikh et al. that closely touch the concept of modus [49–53]. Shaikh designed a graphical user interface that assisted humans in the task of defining paths and objectives for robots during a planar-path planning task. At the heart of this graphical user interface was an "adverb palette," which was used like an artist's palette to mix and match objectives. The adverb options for qualifying the robot motion were: safely, stealthily, and quickly. This concept of robots estimating human intent and modifying their behavior to match the intent is fundamental to our research. In their work, the adverb weights were explicitly set by the human, but in our work we attempt to automatically detect the modus of a team.

The work of Medina et al. considered the effect of language on task execution [34]. They broke down communication that robots would need to detect into verbs and adverbs. In order to test their theories they had participants interact with a haptic-interface robot to navigate paths exhibiting three different adverbials. The adverbials they chose were fast and slow, with an implicit third adverbial of normal when no explicit adverbial was given. They found that participants completed the tasks at different speeds and with slightly different paths when different adverbials were mandated. The fact that human performance changed when different adverbs were specified implies that robots need to detect this intent in order to perform optimally as co-manipulation partners.

2.2 Human-Human and Human-Robot Co-Manipulation

The majority of co-manipulation research is focused on robot algorithm development in the pursuit of making robots better co-manipulation partners for humans. However, human-human studies are an important stepping stone toward better human-robot interaction and many human-robot studies are preceded by human-human studies. Below we outline important work related to human-human explorations and experiments first, then include related literature on the development of human-robot co-manipulation algorithms (whether they are directly built on human-human studies or not). A general survey of related physical human-robot collaboration can be found in [44], which may be useful as an introduction to this broad area.

It is important to note that despite the significant amount of research that has been done around the theme of co-manipulation, there are at least two main aspects that seem to be lacking in the literature. First, the effect of human modi in co-manipulation research is significantly underdeveloped (for both human-only teams and human-robot teams). Second, the number of teams that were manipulating, heavy, large-scale items is quite small and may therefore limit some of the prior results when applied in scenarios, such as search and rescue or disaster response.

2.2.1 Human-Human Co-Manipulation. Townsend et al. researched dyads, or teams of two people, moving through a series of obstacles holding a board outfitted with force and motion sensors and then trained a NN to create a predictive reference trajectory that a robot could use in a control algorithm [59]. Bussy and Agravente et al. also performed a series of studies that started with human-human research and after moved to human-robot research [3, 4, 11, 12]. They also started with co-manipulating an extended object and developing a state machine to model human behavior. In related work by Mielke and Jensen [24, 37], triggers were identified at the start of rotation tasks vs. at the start of lateral translation tasks for large extended objects, suggesting communication signals via forces or torques. Calculating these interactions builds upon the work of Groten et al. [22] where the interaction forces are defined as the smaller of two opposing forces. Noohi et al. also presents an alternative approach assuming minimum jerk movements [39, 42].

Researchers have been able to further classify the states and behavior of human teams co-manipulating extended objects. While physical co-manipulating, Lanini et al. [31] classified walking direction states, Kucukyilmaz et al. [30] classified six behavior states, Al-Saadi et al. [6] successfully classified the actions into four states (comparing harmonious/conflicted translation or rotation), and Karayiannidis et al. [26] explored classifiers to resolve the rotation vs. translation co-manipulation ambiguity and predict human intent in the plane of motion.

Complementary research by Sawers et al. [48] found that interaction forces are critical signals for communication in cooperative motion tasks, such as in dancing with a partner. Further, as human co-manipulation interactions are inherently multi-modal and hard to model, the work of Rysbek et al. attempts to capture the multi-modal nature of those interactions. They also performed a study in which human teams moved an object around an obstacle together [47]. They were able to identify patterns in low-level physical data that can be interpreted as higher-level motion primitives that communicate motion intent between the team members. They theorized that utilizing the information from these motion primitives in a multi-modal interaction manager, as they created in their previous work [2], would allow a robotic co-manipulation partner to synergize the two disparate forms of communication to estimate human intent more successfully.

2.2.2 Human-Robot Co-Manipulation. Although we do not show explicit human-robot co-manipulation control in this article, we believe that understanding prior strategies for human-robot co-manipulation controllers is important to understanding both the novelty and utility of the results presented in this article.

Generally speaking, the majority of prior research in dyadic human-robot co-manipulation uses a form of admittance control (see [15, 17, 19, 35, 36, 58]) or impedance control (see [3, 4, 11, 12, 25, 63, 65]). However, there are many successful general force or motion-based strategies that instead use force or motion to estimate the intent of a human partner (see [5, 7, 8, 28, 32, 57]). There are also learning-based strategies that attempt to either interpret human intention explicitly (see [20, 21, 35, 36, 56, 61, 64]) or to learn a controller directly for human-robot co-manipulation (see [16]). In addition to the specific areas outlined above, a survey article covering methods in both impedance variation and learning strategies in relevant human-robot interaction scenarios can be found in [54].

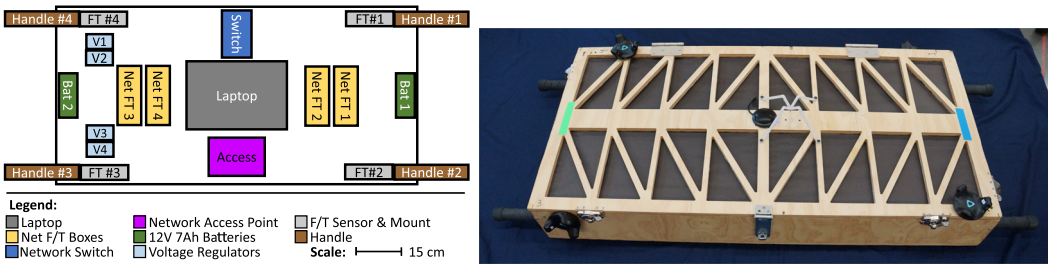

Fig. 1. Diagram of the placement of the different devices inside the co-manipulated object which allowed it to provide untethered power to the force torque sensors and record the resulting data locally on a computer inside the object.

Importantly, the past research in this entire area of pHRI for co-manipulation does not clearly define high or low-performing groups for a well-specified task or variety of tasks (other than perhaps time or force efficiency). Instead, the literature tends to focus on the general potential for success or failure, which is a difficult signal for a robot to operate on since it is not smooth or continuous. In addition, the notion of performing a task with different objectives or weightings on multiple objectives for co-manipulation is not explored. We believe that this is due, in part, to a lack of clearly defined modi during the tasks that currently exist in the literature. Our research attempts to build on past work in the hope of providing data to determine better ways to evaluate a co-manipulation partner's performance, thereby quantifying how to encode and evaluate the performance of a robot partner in real time in future work.

3 Experiment Design

In order to determine if the modus of a team can be identified by externally observable data we designed an experiment to capture data that we hypothesized would be relevant. The experiment consisted of having two participants (together referred to as a dyad or team) move a 27 kg, 1.2 m long stretcher-like object (seen in Figure 1) through a series of five obstacles. A team is asked to perform with one of four different modi for each series of obstacles with multiple repetitions in a randomized order. After developing a NN classifier, a modified version of the same experiment was performed with real-time modus classification.

3.1 Modi Selected

In the ensuing descriptions, it is useful to think of a modus like an adverb. While a verb describes what action is being taken, an adverb's job is to qualify the manner in which the action is taken; it does not describe the action itself. While a team may move in any direction or around any obstacle, the modus is not concerned with the action that is performed but rather the manner in which that action is performed. Human variability implies a nearly infinite number of different possible modi, forming an almost infinite "modus space." A specific design decision was made to only consider the fraction of the "modus space" that are specifically useful for team co-manipulation. More than 20 modi were initially considered but eventually excluded from our experiment, including: gently, quietly, recklessly, lethargically, slowly, angrily, and many more. The main reason these and other adverbs or modi were removed from candidacy is that they did not cross the value threshold of how we would want to design future robots. For example, no robot should act recklessly or angrily while co-manipulating with humans. Similarly, a robot that was too slow in any action would be unacceptable. A secondary reason for dismissal includes highly correlated adverbs which are essentially synonyms, such as gently and peacefully. The

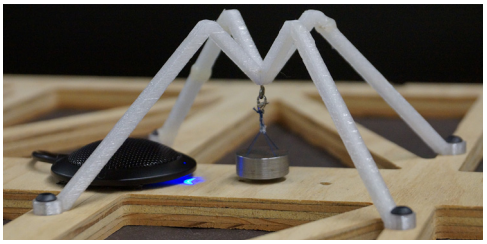

Fig. 2. The pendulum placed on top of the object that the participants carried around the obstacles. The pendulum was placed there for the *smoothly* tasks only.

objective was not to find a single and unique set of modi, but instead to find a spanning set that could represent the different behaviors we expect to see while a team successfully completes a co-manipulation task in a given context. In this sense, this was a design problem that included some subjectivity, but was based on "engineering judgment" which is required for tasks in an undefined area that requires creativity, but is nonetheless grounded in math, physics, and so on (see [45]). This exercise was similar to that used to help define the series of obstacles described in Section 3.3.

The final set we chose was comprised of four modi: quickly, careful of the object, careful of the environment, and normally. The two "careful" modi also help show why an adverb may not be sufficient on its own to describe the desired intent. These modi carry some information about the context as well. Collectively, these four modi do not cover all of the possible modus space, but they cover a sufficiently large part of the useful modus space for co-manipulation in teams. This is especially true since we expect that at times, teams may use a mixture of these objectives to accomplish real-world tasks (i.e., moving a patient quickly but also smoothly is a common goal for medical personnel).

For the purposes of the experiment, example situations or contexts were developed and described to the participants in order to evoke certain behavior as much as possible. When the *quickly* modus was the condition, the participants were told that their friend was injured, and that the friend was safely secured to a stretcher that they needed to move to seek medical attention as soon as possible. For the "careful of environment" modus, participants were told that they worked for a moving company and would get in trouble with their boss if they touched any walls or obstacles with the object. We refer to this modus as *avoiding obstacles* to be more concise.

For the "careful of the object" modus, participants were told that they were bringing a bomb to a safe location where it could be disarmed. The fictional bomb was sensitive to motion, so they needed to move smoothly and carefully to avoid setting the bomb off. In order to help the dyads see how they were moving, we attached a small pendulum to the top of the object they carried and told them that it served as an indicator of their smoothness (see Figure 2). We refer to this modus during the experiment as *smoothly*. The use of a pendulum as an external objective in a co-manipulation task is not unlike the work of Mojtahedi who used a level in co-manipulation tasks between two humans [38].

For the "normal" context, the participants were told that none of the other contexts were invoked and that they should move in whatever manner felt most normal for their dyad. We refer to this modus as *no context*.

3.2 Co-Manipulation Object (CMO) Design

The object that was carried through the obstacles was a wooden box (1.2 m × 0.6 m × 0.17 m). This box will be referred to as the CMO for short. The CMO was designed with the intent to hide sensors and electronics from the participants as much as possible so that they would be more likely to treat the CMO in a natural manner, rather than as an expensive piece of equipment.

The outside of the CMO was outfitted with four HTC Vive trackers that tracked the pose of the CMO as it moved through the obstacles. Four trackers were used to account for possible data dropouts or tracker occlusions by the obstacles or participants. Post-experiment analysis showed that one or more of the trackers was always reporting data at any point in time. These trackers reported pose information at 200 Hz in the form of x, y, z position data and qx, qy, qz, qw quaternion orientation data. The Vive trackers relayed their data over Bluetooth to a computer which ran the Vive software and recorded the data.

The CMO was also outfitted with four force/torque ATI Mini45 sensors, each at the point where one of the stretcher-like handles connected with the CMO. The sensors collect force in the x, y, and z directions and torque in the x, y, and z directions at 200 Hz. In order to store the data without signal loss over a wireless connection, the force/torque data was stored locally on a laptop strapped to the CMO as can be seen in Figure 1.

In order to synchronize the timestamp data between the Vive computers and the laptop storing the force/torque data, we used the "Chrony" daemon for Linux. The devices inside the CMO were placed on axes of symmetry in order to bring the products of inertia of the box close to zero. One conference microphone was placed on the top-center of the CMO which recorded audio from both members of the dyad. All Vive, force/torque, and audio data was published and recorded using the **robot operating system (ROS)** [46].

3.3 Obstacles

We designed varied obstacles with the intent of encouraging different types of motion from the participants. Each of the obstacles specializes in forcing the team to move the CMO in at least one of the six basic degrees of freedom for a rigid body (translation along the x, y, and z axes and rotation about the x, y, and z axes). The desired trajectories for these obstacles all start and stop in the same nominal location in the center of the room, as seen in Figure 3. Although we herein describe the obstacles in a specified order, the participants traversed the obstacles in a randomized order.

The first obstacle was the straightaway, in which the team walked straight to the first location from the starting point, then walked straight for 5 m in the opposite direction, and then moved back to the nominal starting point (shown with numbers 1–4 on the left in Figure 3). This obstacle required only one degree of freedom along the straightaway and covered the most common form of motion, in which a dyad was not impeded by an obstacle and only needed to move straight from point A to point B.

The second obstacle was the hallway, in which the team moved away from the nominal starting point and aligned the CMO in order to move through a hallway (which also had a step) before returning to the starting point. The step was 0.06 m high and induced a pitch rotation of the CMO (about the green axis shown in Figure 4).

The third obstacle was a box obstacle for which the teams lined up with the obstacle, lifted the CMO over the box, and then returned to the nominal starting point. The box usually induced a sidestepping motion from the participants, as well as a lifting motion in order to raise the object over the box. The box obstacle required two of the six basic degrees of freedom possible for rigid-body motion.

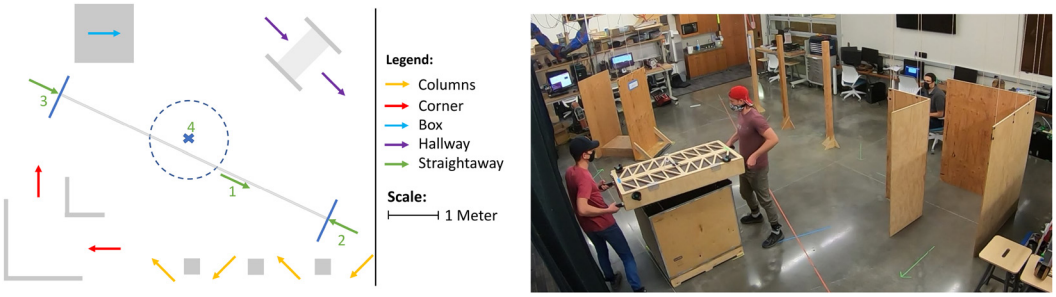

Fig. 3. Layout of the obstacles in the room where the experiment took place. The arrows indicate the direction in which we instructed participants to navigate through the obstacles. Traversing each obstacle involved leaving the blue circle with a blue "x," following the arrows before returning to the blue circle. The straightaway required walking through the blue circle on the long stretch. For orientating the reader, the corner obstacle shown on the far right of the image, corresponds to the obstacle profile on the bottom left of the diagram with red arrows.

The fourth obstacle was the columns obstacle, in which the participants had to weave their way around three columns which were meant to cause a turning motion in yaw of the coordinate frame of the CMO. If the dyads traversed the columns by pivoting around each column, they totaled 520° of rotation, but participants were not constrained to traverse the columns in this way. The last obstacle was the corner, which was spaced so that the participants could only move the object they were carrying through if they rolled the object 90° about the red axis of the CMO shown in Figure 4). Video of the dyads moving through the obstacles was captured using ceiling-mounted cameras. An image from the perspective of one of these cameras can be seen in Figure 3.

3.4 Experiment Procedures

The participants were informed in advance that they would be teamed with someone of a similar height. We thus matched participants whose height difference was less than 0.15 m. All the participants were adults between the ages of 18 and 40. Eighteen of the teams were male and twelve of the teams were female. At the beginning of the experiment, participants were shown an introductory video and completed the necessary articlework. They were then introduced to the obstacle course with a prepared script. After that, the experiment administrator instructed the participants on how to place ankle trackers on their legs properly. Data collection then began. To sync the data from the cameras and the microphone on the CMO a loud clapping noise was made as a reference time in all the recordings.

The main body of the experiment consisted of the participants picking up the CMO, traversing through all five of the obstacles in a random order, putting the CMO down, and then repeating this process with different modi. In total, the dyads completed each obstacle 14 times, for a total of 70 individual tasks completed. For the first two rounds of five obstacles, the participants were not told to perform the tasks in any particular manner (i.e., modus). After the second round was completed the participants were introduced to the different contexts using a prepared script. The participants then did four rounds of five obstacles, each round with one of the four modi invoked. They were then offered a break before four more rounds, followed by another break and four more rounds. In total, they did each obstacle three times with each modus. To provide additional clarity about what the experiment was like, footage from one of the experiments is available at https://youtu.be/eg2mf6LeDlo.

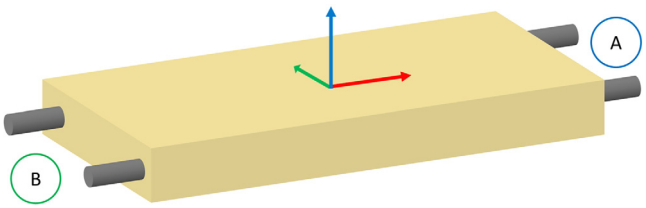

Fig. 4. The table frame is at the center of the CMO. The x-axis is in red and points from Participant B toward Participant A. The y-axis is green and points to the left of the x-axis. The z-axis is blue and points up from the top face of the CMO.

In addition, we have posted the full dataset (before filtering and fusion as described in the following section) at the following link: https://figshare.com/articles/journal_contribution/Classification_of_Comanipulation_Modus_with_Human-Human_Teams_for_Future_Application_to_Human-Robot_Systems/26029339

3.5 Data Cleaning and Fusion

To ensure that the NN learned patterns in physically meaningful signals for effective classification of modus, the data post-processing included data cleaning, filtering, outlier removal, and so on. Since NNs act as universal function approximators, and as such they have the potential to learn any and all patterns in the training data, our careful filtering system was intended to remove any patterns in noise that the NN may learn, ensuring that the NN learns useful patterns in the true signals instead.

The data was manually cleaned with a custom Python application which allowed researchers to label the start and stop of each task. This labeling was not essential to our results presented in this article on classifying modus but makes future analysis and comparison easier between each task, dyad, and obstacle type. Each labeling was also verified by a second researcher. The pose signals from the Vive system would occasionally drift (beyond their nominal noise characteristics) due to occlusions. These data errors were obvious to identify and were manually removed from the dataset.

In order to combine the pose measurements from the four Vive trackers they were first re-sampled and interpolated in time to be at the exact same timesteps. The pose signals were then transformed to give their estimates of the pose of the center of mass of the CMO. The pose measurements were then combined by taking the weighted average of the poses. The Cartesian position was averaged with simple weighted averaging. The average orientation quaternion was found by taking the eigenvector corresponding to the largest eigenvalue of matrix Q as calculated in Equation (1), with subscript i iterating over the quaternions from the four trackers. This has been shown to be a robust method of averaging quaternions [33]. The weights were determined by the Euclidean distance from the direct average of the poses

$$\mathbf{Q} = \sum_{1=1}^{4} \hat{w}_i \mathbf{q}_i \mathbf{q}_i^T. \tag{1}$$

The orientation chosen for the center of mass can be seen in Figure 4. This frame was chosen to be consistent with previous work [24].

The numerical derivative of the pose signal was taken to obtain the velocity signal. The quaternion derivative was converted to an angular velocity $\boldsymbol{\omega}$ using Equation (2). This transformation also changes the frame to the body frame (fixed to the CMO) which is more appropriate for a velocity signal. The $*$ symbol represents the Hamiltonian product, and the IM operator returns the imaginary

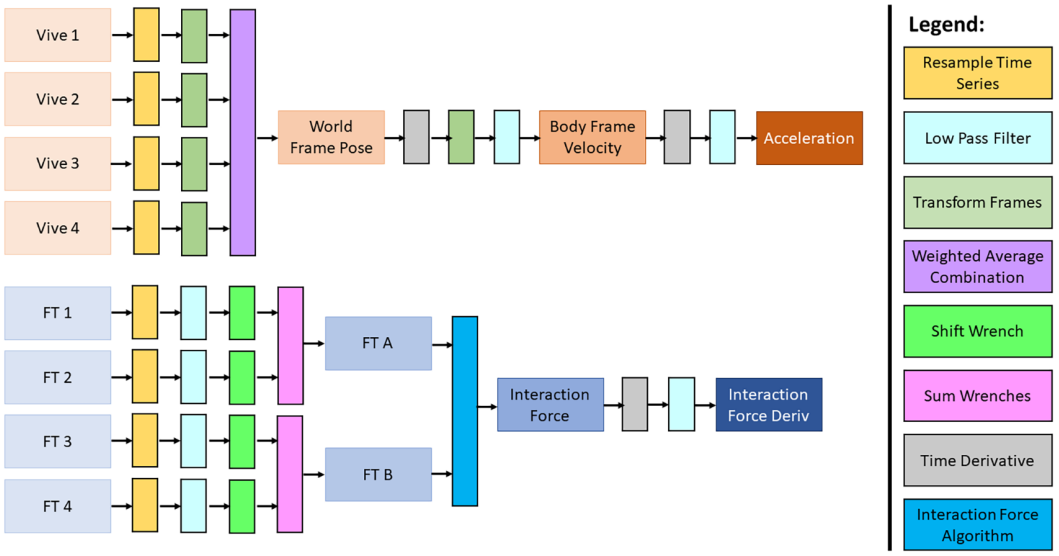

Fig. 5. A depiction of the different procedures used in each step of data processing, showing how all of the vital signals for analysis were compiled.

i, j, k parts of the output of the Hamiltonian product

$$\boldsymbol{\omega} = Im(2\mathbf{q} * \dot{\mathbf{q}}). \tag{2}$$

The velocity signal was filtered with a low-pass filter which had a pass-band frequency of 10 Hz, a stop-band frequency of 20 Hz and 80 dB of attenuation by the stop-band frequency. The pass-band of 10 Hz was chosen because fast voluntary human movement has a frequency of about 3–5 Hz [10]. We wanted to capture these frequencies and reject any noise from the sensors that recorded data at 200 Hz. This same filter design was used to filter all of the other signals.

In order to produce the acceleration trajectory signals the derivative of the velocity signals were taken and then filtered again. The derivative was taken using the central difference of sixth-order accuracy as seen in Equation (3). The subscripts in Equation (3) represent the time at which the x value was sampled. The values added or subtracted from the subscripts represent indexing forwards or backward in the sample

$$F'(\mathbf{x}_i) = -\frac{\mathbf{x}_{i-3}}{60} + \frac{\mathbf{x}_{i-2}}{20} - \frac{\mathbf{x}_{i-1}}{4} + \frac{\mathbf{x}_{i+1}}{4} - \frac{\mathbf{x}_{i+2}}{20} + \frac{\mathbf{x}_{i+3}}{60}. \tag{3}$$

The larger central difference kernel has a smoothing effect. The flow of data and mathematical procedures used to produce all these signals is summarized in Figure 5.

In order to find the interaction wrench between participants, the combined wrench (meaning the sum of torques and forces from each handle for a given participant) had to first be determined. The raw wrench signals from each force-torque sensor were re-sampled and linearly interpolated in time to have the exact same timesteps. Each wrench signal was then filtered with a low pass filter. Then the wrenches were shifted in pairs from the sensor locations in the handles to be directly in front of the participants and halfway between the handles. The shifting was performed using the shifting law seen in Equation (4). In Equation (4), i and j represent frames one and two of the transformation, and bc indicates that the transformation moves the wrench from point b to point c in space. The matrices are 6×6, each sub-matrix being a 3×3. The zeros represent 3×3 sub-matrices of zeros and the $\mathbf{I}$ represents a 3×3 identity matrix. The $-[\mathbf{r}^i_{bc}]_\times$ term represents

forming a skew-symmetric matrix from the vector pointing from point b toward point c in frame i

$$\mathbf{Z}_i^j(bc) = \begin{bmatrix} \mathbf{R}_i^j & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_i^j \end{bmatrix} \begin{bmatrix} \mathbf{I} & -[\,\mathbf{r}_{bc}^i]_\times \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{4}$$

The summation of the two wrenches between the handles most accurately represents the combined wrench exerted on the box by one participant. These individual participant wrenches were then combined using the interaction force calculation algorithm implemented by Jensen and Groten, to calculate the interaction between the two participants [22, 24]. A more detailed explanation of this algorithm can be found in their work, specifically under the heading "algorithm 1." Here we include only a brief summary of the algorithm.

If two forces or torques act in the same direction then there is not an interaction force or torque in that direction. In other words, the combination of the forces or torques of the two individuals are working together to move the object. Interaction forces and torques occur when forces in excess of those required to move the CMO are applied either for stability or in order to communicate haptic information with the other person in the dyad.

Interaction forces and torques can then be found when two force or torque signals act in the same direction but have opposite signs. In this case, one participant is applying more force or torque, enough to both move the object and counteract the force and torque from their partner. The magnitude of the smaller of these two signals is then considered the interaction force or torque, since it represents the magnitude of force or torque that each participant is exerting that does not accelerate the CMO. The interaction wrench signal often has portions of the signal with values flat-lined at zero. These portions of the signal happen when the dyad is working together to move the CMO with no excess energy expended as an interaction force or torque.

The derivative of the interaction wrench was also taken with the central difference of sixth-order accuracy. This derivative signal was then filtered to produce the final signal used in the offline analysis. This signal is simply referred to as the derivative of the interaction wrench.

3.6 Real-Time Experiment

Once we had trained the NN (as described in Section 4.1) on the data that was cleaned offline, we adapted our system to work in real time and performed another series of human-human co-manipulation experiments. Adaption only required the forward-backward filtering and data fusion that had been done offline to be switched to real-time filtering or data processing. We also dropped the steps related to outlier rejection which were done more easily offline.

The real-time NN is meant as a validation of this algorithm's potential for real-time application with a human-robot dyad (instead of a human-human dyad). For this reason, we used a simpler version of the NN trained only on velocity data. The velocity of the co-manipulated object was chosen as a signal because it can easily be estimated from the velocity of the robot base and end effector, or with basic computer vision. Since the force-torque sensors were not required for the real-time study, they were removed from the table, along with the onboard laptop. Without these components, the table was about 4.5 kilograms (or 17%) lighter. The simplified real-time experiment had three obstacles instead five. The three obstacles were the columns and straightaway described in Section 3.3, as well as a new obstacle called the "ring" to test how well the NN generalized to new obstacles and motions. The ring consisted of a single column, which the participants had to walk past and around before returning to the start position.

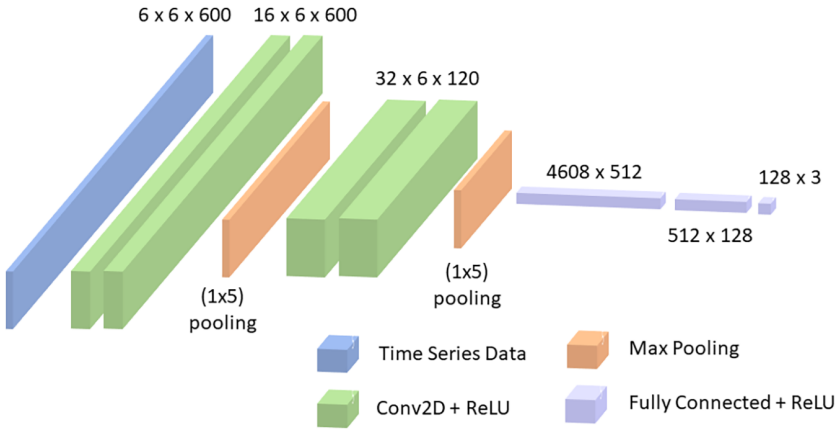

Fig. 6. The architecture of the CNN used to classify the modi as the participants performed the tasks of the experiment.

4 Results

4.1 Offline Results

We developed a **convolutional NN (CNN)** and trained it on the data from the experiments. This NN attempts to determine the modus of any given team at any given time based on the data that a team produces as they move. The following signals were used as inputs for the NN: velocity (of CMO), acceleration (of CMO), wrench from participant A, wrench from participant B (as labeled in Figure 4), interaction wrench, and the derivative of the interaction wrench. These signals all have six components (x, y, z linear and x, y, z angular), making a total of 36 time-series signals to be used in the classification.

The data from all 30 teams was split into subsets of training and validation data. The training data consisted of the data from 24 of the 30 teams with the validation data comprised of the data from the remaining 6 teams. All obstacles were used in both training and validation to make the classifier more robust. A sliding window of 3 seconds was used as the time frame over which classification was attempted. At 200 hertz, 3 seconds results in 600 timesteps worth of data as input to the NN.

The NN architecture was based on the VGG 16 architecture, but was modified for this application [55]. All 36 signals were independently normalized in time by finding the minimums and maximums of the each signal across the whole study and scaling the training data so that the maximum and minimum of each signal were 1 and -1, respectively. Instead of stacking all 36 signals on top of each other in a $1 \times 36 \times 600$ tensor, the different categories (velocity, acceleration, and so on) were treated as different channels in an image, so the tensor dimensions were $6 \times 6 \times 600$. The data was fed through two convolutional layers, a max pooling layer, two more convolutional layers, one more max pooling layer, then through three linear layers and a final soft-maxed operation to return classification confidence values. Dropout layers were included in training, but not in validation. Importantly the max pooling was only in the time dimension, since we did not want to lose data across different Cartesian directions by max-pooling the columns. A visualization of the NN design can be seen in Figure 6.

The choice to use a modified VGG 16 architecture was motivated by two major considerations. First, we knew our goal was classification and VGG 16 performs especially well in classification, specifically image classification. Second, CNNs like VGG 16 do well at learning spatial relationships.

Table 1. Performance of the Four-Modus NN over 10 Different Cross Validations

Cross Validation Number	Avoiding Obstacles Accuracy%	Quickly Accuracy%	Smoothly Accuracy%	No Context Accuracy%
1	61.2	51.2	89.9	25.2
2	48.3	78.2	98.8	32.5
3	38.4	79.9	91.2	43.6
4	11.5	84.2	93.7	76.0
5	44.2	71.5	94.2	51.3
6	17.2	91.9	97.2	65.0
7	37.6	71.3	94.4	50.6
8	27.6	81.9	95.7	61.1
9	32.8	47.3	91.2	51.9
10	31.6	64.3	97.5	53.2

By formatting our time-series data like an image, with adjacent timesteps becoming adjacent "pixels" in the image, we translated the time-series relationship of data points into a spatial relationship which the NN could extract.

Other researchers, such as Zhao et al. and Chen et al. have also used CNNs for classification of time-series data, so there is a precedent for adapting a CNN for such tasks [13, 43]. We chose to modify the architecture, making it smaller because our dataset had much less training data to work with than VGG 16 and we only wanted to classify across three or four classes instead of 1,000 in the original VGG 16 architecture [55].

Throughout the course of the experiments, we noticed that the behavior of the teams for the *no context* modus and the *avoiding obstacles* modus were difficult to distinguish from each other from a strictly observational standpoint. In order to determine if *avoiding obstacles* and *no context* were clearly separable, the analysis of the NN was performed twice to compare a NN with an output of three classes to a NN with an output of four classes.

In order to robustly characterize the performance of the NN, cross-validation was performed. Each row in Tables 1 and 2 represents a different division of all the data into sets of 24 training teams and six validation teams. Cross-validation helps show if a specific split of the data was particularly high-performing or not. Average performance across all splits is an important measure of robustness, but it is also common practice to choose the best NN for future work. Classification accuracies were calculated by randomly selecting 3 second time windows from the validation data. Each time window results in one classification from the NN and time windows may overlap. For obstacle-specific accuracies, 20,400 time windows (and thus 20,400 samples) were used to calculate the average accuracy of correct classification. The number of samples used was defined by increasing the number of windows until stability of results was observed. For the overall accuracies of each NN, 54,000 samples were used to reach stability (with stability defined as accuracy results differing by 0.25% or less across multiple runs of the accuracy analysis for the same NN). The first classifications were delayed until 600 timesteps after the start of the tasks because 600 timesteps of data is the selected input size for the NN.

As can be seen in Table 1, the NN classified between four modi with a total classification accuracy that varied between 56% to 68%, the average accuracy being 63%. The *quickly* and *smoothly* modi were classified most successfully with average classification accuracies of 72% and 94%,

Table 2. Performance of the Three-Modus NN over 10 Different Cross Validations

Cross Validation Number	Avoiding Obstacles Accuracy%	Quickly Accuracy%	Smoothly Accuracy%
1	77.6	64.1	95.1
2	85.6	80.3	92.2
3	68.9	89.5	94.8
4	84.7	72.0	95.5
5	86.6	74.1	95.3
6	71.1	91.6	97.9
7	79.6	78.0	92.5
8	74.2	95.0	94.3
9	64.8	71.6	90.7
10	78.1	80.7	94.0

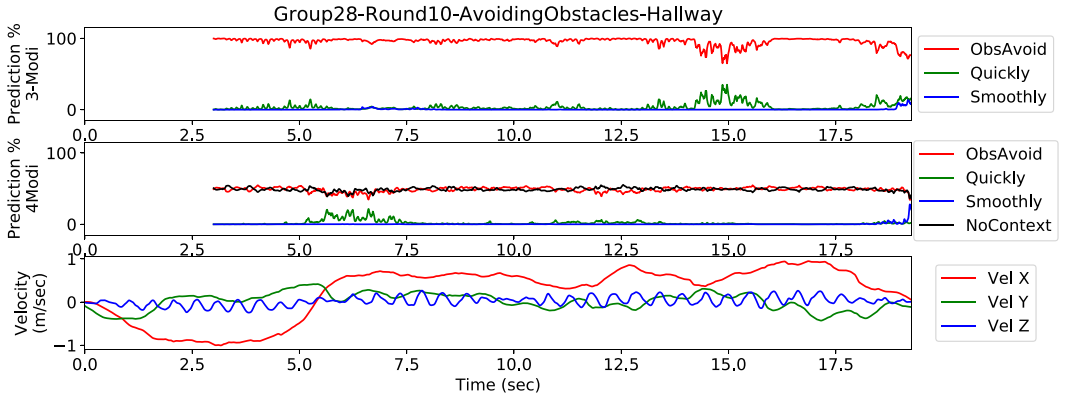

Fig. 7. Performance of the three-modus and four-modus NNs for one task from team 28. The soft-max classification estimates from the NN are seen above, and the linear velocity xyz signals are shown below to give some idea of what is happening in the task. The classification does not start until 3 seconds (600 time steps) into the task since that is the size of the classification window.

respectively. The *avoiding obstacles* and *no context* modi were classified less successfully with average classification accuracies of 35% and 51%, respectively. As mentioned previously it was difficult to determine any difference in the behavior of teams between *avoiding obstacles* and *no context* suggesting a default human behavior may be to avoid obstacles in the absence of any other provided objective.

The data in Figure 7 helps to support the hypothesis that *avoiding obstacles* and *no context* are similar. We see that the four-modus NN split the difference between the *avoiding obstacles* and *no context* modi causing classification performance to be low on both of these tasks as the classification kept oscillating between the two modi. For the four-modus test shown in Figure 7, the NN achieved a classification accuracy of only 55% across this task. Alternatively, the three-modus NN achieved 100% classification accuracy across the whole task. We do not hypothesize that *avoiding obstacles* and *no context* are wholly indistinguishable, but rather that the behaviors are so similar that it is

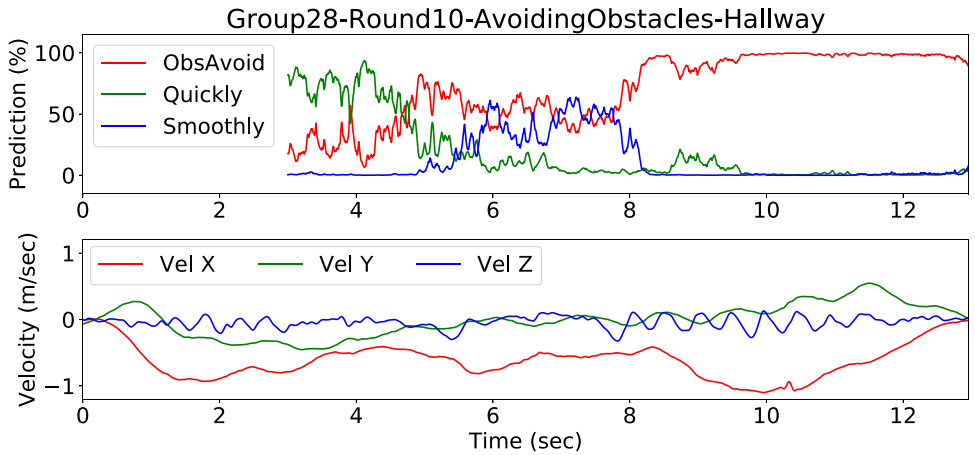

Fig. 8. Performance of the three-modus NN on one trial of data, in which the NN struggled to consistently classify the modus of the team. The soft-max classification estimates from the NN are seen above, and the linear velocity xyz signals are shown below to give some idea of what is happening in the task. The classification does not start until 3 seconds (600 time steps) into the task since that is the size of the classification window.

not beneficial to differentiate them for the sake of a robot controller. We therefore made a design decision to primarily use the three-modus NN.

The results from the three-modus NN can be seen in Table 2. The total classification accuracy varied from 76% to 89%, with the average accuracy being 84%. While this is significantly better than in the four-modus NN, an increase in classification accuracy is inevitable when there are fewer classes. However, the most important improvement in using the three-modus NN over the four-modus NN is that we no longer see the behavior of an oscillating classification between two modi.

In Figure 8 the NN classified this task with 74% accuracy over the entire task. This trial shows that at different points throughout the task the modus of the team was estimated to be *smoothly*, then *avoiding obstacles*, then *quickly*, then back to *avoiding obstacles*. While this could be a failing of the NN to classify the behavior of the team, it also hints at the possibility that teams were not always consistent in the execution of tasks in certain modi. Figure 8 may be showing that the team was altering their behavior throughout the task. We see that the team moved a little more quickly to the obstacle, moved a little more carefully through the obstacle, and then had a more relaxed pace back to the starting point.

It is important to remember that we have no ground truth data, other than encouraging participants as much as possible to move in a certain manner. Rather than a failing of the NN, this transition uncertainty might show the ability of the NN to determine variations from baseline modi behavior and to identify transitions from one modus to another. One of the benefits of the soft-max result is that it shows the ratios and trends of confidence. The softmax can predict a change in the predominant modus before it happens as the confidence values change. Additionally, the modus of a team might not always be exactly in one class, but might be a weighted combination of multiple classes, just like in work of Shaikh et al. [49]. Watching the trends of all three values can help predict a change in modus as a team shifts from one modus to another in certain situations. This is one of the most important lessons to learn from this research to apply to robotics. Not only can a NN classify the modus of a team with a high degree of accuracy, it can also predict changes in modus by observing the confidence levels of each modus. The ability to detect modus, assert

Table 3. The Confusion Matrix for the 3-Modus NN Trained on All Signals

	ObsAvoid (Predicted)	Quickly (Predicted)	Smoothly (Predicted)
ObsAvoid (True)	73.0%	18.6%	8.4%
Quickly (True)	5.2%	94.7%	0.1%
Smoothly (True)	6.0%	0.0%	94.1%

Table 4. The Confusion Matrix for the 4-Modus NN Trained on All Signals

	ObsAvoid (Predicted)	Quickly (Predicted)	Smoothly (Predicted)	No Context (Predicted)
ObsAvoid (True)	44.2%	6.4%	4.2%	45.1
Quickly (True)	8.2%	71.5%	0.5%	19.8
Smoothly (True)	5.5%	0.1%	94.2%	0.2
NoContext (True)	38.1%	8.1%	2.5%	51.3

Table 5. The Mean Classification Accuracies for Each Obstacle-Modus Combination from the Validation Set Evaluated with a Sample of 20,400 Classifications of the Validation Data Per Obstacle

	ObsAvoid (mean)	Quickly (mean)	Smoothly (mean)
Box	69.3	95.2	85.7
Columns	70.1	98.1	89.1
Corner	80.5	90.9	94.6
Hallway	69.6	93.7	97.5
Straight	75.9	95.1	95.5

confidence in those predictions, and predict changes in modus can go a long way in improving the reactivity and helpfulness of robot co-manipulation team members.

Tables 3 and 4 show the confusion matrices for the three-modus and four-modus NNs. In this table, we see that the NNs struggled to classify the *avoiding obstacles* modus most often and had the easiest time classifying the *smoothly* modus. The *smoothly* modus was the most distinct of the three with the slowest speed and careful motion, so it makes sense that it was the easiest to classify. The *avoiding obstacles* modus was in between the other two modi and was the easiest to misclassify if the team had a particularly fast or particularly slow baseline speed. Table 5 shows the average classification accuracies of each obstacle-modus combination. The box obstacle was the hardest to classify. This may be because of the sidestep motion it induced and that was not normally seen for other obstacles. The corner and straightaway obstacles were the easiest to classify. However, all the classifications performed relatively well, which is promising for the generalization of the classifier.

We originally hypothesized that of the six signals used in classification, velocity, acceleration, and interaction wrench would be the most important and carry the most information, and thus be the most crucial for the NN classification. In order to test this hypothesis we modified the three-modus NN and trained it multiple times with different combinations of input signals to produce the results shown in Table 6. We were surprised to find that interaction wrench was the second least useful

Table 6. Average Classification Performance of the Three-Modus NNs Trained on Different Combinations of Inputs

Signal Combination	Average Accuracy	Minimum Accuracy	Maximum Accuracy
V	82.8	77.2	88.8
A	82.5	75.5	87.4
FTA	77.7	71.6	81.9
IW	73.7	68.0	80.0
IWD	69.4	33.3	80.0
A/V	84.4	76.1	88.0
A/IW	82.7	75.5	87.0
IWD/V	82.5	76.3	87.6
A/FTA	82.1	73.9	85.9
IW/V	81.8	76.2	85.8
FTA/V	81.3	71.7	86.5
FTA/FTB	78.8	72.6	81.7
A/V/IW	84.3	76.9	88.9
A/FTA/V	83.3	76.5	87.5
A/FTA/FTB	82.7	76.9	87.0
FTA/FTB/IWD	79.4	72.9	82.5
A/V/FTA/FTB	83.6	78.1	86.9
V/A/IW/IWD/FTA/FTB	83.6	76.4	87.1

Double lines between rows separate groups of NNs trained on one signal, two signals, and so on up to six signals. A: 6-dimensional vector of linear and angular acceleration for the CMO; FTA or FTB: 6-dimensional force and torque measurements from participant A or B; IW: 6-dimensional interaction wrench (force and torque in three directions); IWD: time derivative of the interaction wrench; V: 6-dimensional vector of linear and angular velocity for the CMO.

signal, the least useful being the derivative of the interaction wrench. This is likely because at times when the team is working together, the interaction force signal flattens to zero since the team is cooperating with no conflicting interaction. While a flat portion of the signal does provide some information, it is a feature-poor signal for the NN to analyze. Despite the unexpected result, this is promising news for future research, as it shows that easy-to-measure signals, such as velocity can perform very well in classification tasks. This lowers the barrier for modus classification in real-time robot co-manipulation control algorithms and allows us to focus on appropriate real-time processing techniques for effective classification using the most promising signals.

Of NNs trained with single signals, the highest performing were those trained with velocity or acceleration signals, which performed with an average accuracy of 82.8% and 82.5%, respectively. This is an excellent outcome for robotic controller design since there are multiple simple ways to acquire quality velocity and acceleration signals for the CMO. NNs with multiple signals generally performed better than those trained with individual signals, confirming the intuition that including more signals results in better performance. It is noteworthy that the all of the NNs which were trained on only velocity and/or acceleration performed better than those nets trained with velocity and force signals or acceleration and force signals. This probably has to do with the suboptimality of our CNN architecture rather than the force signals themselves, but it demonstrates that the

Table 7. The Mean Classification Accuracies of the Real-Time NN on Offline Validation Data and on Real-Time Data, Separated by Modus

	ObsAvoid (mean)	Quickly (mean)	Smoothly (mean)	All Modi (mean)
Clean Data	76.2	92.8	98.5	89.2
Real-Time Data	68.1	95.7	75.3	79.6

Note that these accuracy averages are weighted so each task weighted equally independent of its duration, so these averages differ slightly from those shown in the confusion matrices from Table 9.

Table 8. The Means Classification Accuracies for Each Obstacle-Modus Combination from the Real-Time Experiment (Comparable to Table 5)

	ObsAvoid (mean)	Quickly (mean)	Smoothly (mean)
Columns	68.4	94.5	74.2
Ring	57.2	99.8	76.4
Straight	79.3	93.2	75.3
All Obstacles	68.1	95.7	75.3

interaction between the input data signals and the NN output is not always straightforward and should be considered carefully. If force-torque signals are available to future researchers on the systems they use, then force and torque signals can be considered for modus classification, especially if different classification schemes are being explored. However, given the superior performance of NNs trained only on velocity and acceleration data in our experiment, we recommend that future modus classification efforts focus on those signals first.

4.2 Real-Time Results

For the real-time experiment, we used the highest-performing velocity-only NN trained on the offline data. The real-time classification accuracy was 79.6% overall as compared to the 89.2% validation accuracy with offline data. The real-time classification performed similarly across each obstacle, and its performance varied significantly across modi like the offline classification. The real-time results separated by obstacle and modus are included in Table 8. The overall real-time accuracy compared to the NN's validation accuracy on offline data is shown in Table 7. A video of the real-time classifier (meaning the classification was done while data was published through ROS, and the classification was recorded in real time), can be seen at https://youtu.be/GO41O7npDI8.

We can draw a few important conclusions from these results. First, we note that there is potential for a modus-classifier to perform well on tasks it was not trained on as classification for the quickly and smoothly modi during the ring task was accurate. Second, we can see that although the NN performed similarly between the real time results and the offline results, there is a noticeable difference between the NN's performance with real-time-filtered data and offline data. We attribute this difference to the fact that the NN was trained on offline data rather than real-time filtered data. Intuitively, a noisy signal is less smooth and outliers would cause jumps in the estimated velocity signal. We can see that by comparing the confusion matrices of the real-time and offline performance in Table 9 that *obstacle avoidance* is misclassified as *quickly* much more in the real-time NN and *smoothly* is misclassified as *obstacle avoidance* more with the real-time data than the offline

Table 9. The Confusion Matrix for the NN Used in Real-Time Study, Normalized across the True Modus (Rows)

	ObsAvoid (Predicted)	Quickly (Predicted)	Smoothly (Predicted)
ObsAvoid (True, Real time)	75.1%	19.2%	5.7%
Quickly (True, Real time)	8.4%	91.6%	0.0%
Smoothly (True, Real time)	22.1%	1.6%	76.3%
ObsAvoid (True, Clean Data)	75.1%	16.1%	8.87%
Quickly (True, Clean Data)	7.18%	92.8%	0.00%
Smoothly (True, Clean Data)	1.82%	0.00%	98.2%

The first three rows show the confusion matrix for the NN on real-time data and the final three rows show the confusion matrix for the same NN on the carefully-cleaned validation data.

data. One way to likely improve performance would be to train the real-time classifier directly with "real-time" data that is filtered and processed in the same way as the presented real-time experiment (even if the training occurs offline).

5 Discussion

5.1 Insights on Physical Human-Human Co-Manipulation

The purpose of this work was to answer two important questions. First, when the modus of a team varies, does the behavior of the team change enough to be observable? Second, if the change in behavior with varying modi is observable with certain signals, can those observations be used to classify the modus of the team?

The answers to both of these questions is yes. The behavior of the teams was indeed different when different modi were invoked. This is clear from the observation of the experiment administrators, but much more importantly it has been shown that a computer can detect and classify the difference. The fact that the difference can be detected with up to 89% accuracy is significant. It means that humans are similar enough in how they move that learning from a few teams can generalize fairly well. As a future strategy, we plan to use a NN trained on high-performing teams only as a good starting point for additional refinement and learning with an unknown team, rather than trying to learn that team's modus model from scratch. Another important finding of our work is the similarity between the *no context* and *avoiding obstacles* modi. This seems to indicate that people perform obstacle avoidance as a default behavior.

5.2 Lessons for Physical Human-Robot Co-Manipulation

In this section we describe lessons, necessary considerations, and future strategies to enable applications of our modus classifier to human-robot co-manipulation controllers.

This research serves to point out a gap in the current field of human-robot co-manipulation research. The tradeoff in competing objectives or methods for completing a task (which we call modus) has not been significantly explored in human-robot co-manipulation research, but it significantly impacts the behavior and performance of human teams. We expect that the presented modus classifications can provide significant direction for human-robot researchers as they identify

explicit parameters to modify robot control algorithms or optimizations that might allow the robots to adapt more appropriately to their human partner's behavior.

The reason that clearly and explicitly defining goal behavior is important (as opposed to defining only the goal state of the co-manipulated object) is to enable more useful and intuitive co-manipulation control. Many robot controllers use optimization in the loop, model predictive control being one of the most well-known. These controllers need clearly defined objective functions to optimize. When a team is behaving according to a different modus, the objective function should change appropriately, or weightings on certain parts of the objective function might need to change. This could lead to an increase in satisfaction from the human partners and make the robots more generally useful or task-efficient.

In addition, since obstacle avoidance seems to be a default behavior for human teams, obstacle detection and obstacle avoidance algorithms should likely be a baseline for robot helpers. Research will need to be performed to explore how to balance a robot's capability to detect the obstacles in its path, while also meeting the desired intent of the human partner as far as both object trajectory and modus are concerned.

Delay in classification is another element that will need to be addressed for real-time application in co-manipulation controllers. This has been addressed to some extent with our real-time implementation and validation. However, the classification window uses 3 seconds of past data, while our real-time filtering adds 0.075 seconds of lag. The NN also only currently runs at 10 Hz, but this is not a fundamental limit with the NN (whose speed is hardware dependent and was run at 200 Hz on a non-specialized desktop computer in a test case). Given these delays, the current worst-case lag would be 3.175 seconds of delay between when a modus or behavior begins and the moment when a robot could recognize the change. However, this is the worst case. We expect that even for a human teammate to recognize a change in behavior, it may take 1–2 seconds. Given the natural delay between two human partners, it may be that the 3.175 seconds of delay would be acceptable between a human and robot partner. Additional exploration of the NN's sensitivity to filtering parameters and numbers of inputs (by varying the amount of past data used) may also yield improved real-time performance with lower amounts of lag. Although we have not done extensive testing to determine the sensitivity of our algorithm to changes in modus, we expect this not to be a significant problem, given that most tasks can last minutes without changing modus. This results in our 3.175 second worst-case-delay being orders of magnitude less than the time for which the modus would be relevant during operation.

Finally, during the time period when the robot is waiting for enough data to determine the modus with which to operate, the robot would need a default behavior or cost function. Based on our results, we would propose setting the default behavior or modus as *obstacle avoidance* since that was the closest to the *no context* or neutral motion than any of the other modi. After the initial lag, the robot would then adapt the cost function it uses according to the current estimate of the modus. Since the estimate will always lag real or desired behavior to some extent, the robot would only change its active behavior when the newly classified modus was maintained for at least the duration of the lag. More practically, we expect to not use discrete classification signals to determine co-manipulation behaviors or cost functions, but instead to use a relative weighting between three possible behaviors as shown with the continuous values reported in Figure 8. We hypothesize that this approach is more aligned with how human agents behave as they may have multiple objectives that vary in importance based on circumstance or environment. In this way, robot behavior could also change smoothly as the output of the NN changes in time. By way of example, the robot may switch from a cost function which maximizes speed if the modus is quickly, to a cost function which minimizes jerk if the modus is smoothly, to an obstacle avoidance cost function if the modus is obstacle avoidance, or to a multi-objective cost that includes weighted versions of the previous three deriving from soft-max classification estimates.

5.3 Limitations and Future Work

In this section, we first outline a number of limiting factors related to our study formulation and execution. We next discuss future work that will be necessary to eventually enable realistic human-robot co-manipulation scenarios.

One limitation of our experiment was the lack of consistency in modus execution of any given team throughout their experiment. There is no guarantee that when we instructed teams to move in the manner of one of the modi, they actually moved only in that modus for the entire task, or matched their behavior for repetitions of that task. There was also a lack of consistency between the teams. While all of the teams generally moved faster on the *quickly* tasks and slower and smoother on the *smoothly* tasks, they did not all do so equally. Some teams had a faster baseline speed than other teams. In the future, fine-tuning the NN for each new team during co-manipulation could substantially improve the performance of the NN by effectively biasing it to the baseline behavior of any particular team. This practice could also be used for human-robot teams, in which the robot would spend a few minutes, during initial operation, to calibrate its NN and to adjust to the baseline of its human partner. This would also help account for variance in team behaviors from the average behavior.

From our observations, some dyads were able to perform at a higher level than other dyads. They were either faster, smoother, had less conflict, or had less confusion. What makes the difference between these teams is still unknown. If a robust manner of identifying high-performing teams could be identified, then these teams could become models for robot controllers to mimic. Future analysis of this data should include the identification of high-performing dyads. Our data may also include insights into human fatigue. The teams in the experiment co-manipulated a heavy object for close to half an hour, and fatigue definitely played a role in certain behaviors, especially for the teams that were less physically conditioned.

Only one CMO was used. This object never varied significantly in weight or dimension (aside from the 17% change for our real-time experiment). People almost certainly change their behavior with heavier/lighter or smaller/bulkier objects. We also told participants to hold the CMO in a very specific way with one hand on each handle at all times. This prevented body contact with the object in order to support some of the weight. It also prevented choosing another grip style, which might affect behavior. However, the result that a velocity-only network gave reasonable performance in real time may mean that when using the right signals, people's strategy for interacting with the object become less important.

We also do not claim to have found the optimal classifier for modus. While the architecture was designed based on best practices, there are many different classification algorithms and frameworks that could be applied to the same problem, some of which are specifically tailored to time-series data. Residual networks and recurrent NNs, for example, are commonly used for time-series data. Additionally, there are classification methods that specifically consider uncertainty and multimodality (many possible distinct futures) in their estimates, from which our classification could benefit. Bouveyron and Girard use a Gaussian Mixture Model to explicitly consider uncertainty in a classification task [9] and Choi et al. use a mixture density network to consider uncertainty when determining how a self-driving car should behave [14]. Ivanovic et al. use a conditional variational autoencoder with long-short term subcomponents to consider multiple future possibilities, which we could adapt to consider several different ways in which the modus may change in a co-manipulation task [23]. It is possible that another algorithm would have performed better on the data, or that a different NN architecture could have achieved a better performance. Similarly, we did no analysis of time windows used for classification. Accurate classification is likely possible with time windows shorter than 3 seconds, and the optimal time likely differs between classifier architectures. However,

our goal to prove that a modus is distinguishable and classifiable in real-time was achieved and the performance in this article can act as a benchmark or baseline for future work.

6 Conclusion

This research has shown that when the desired modus of a team varies, the behavior of the team changes enough to be observable. It has also been shown that the proposed measurements (i.e., velocity, force, and so on) can be used to classify the modus of the team using NNs both offline and in real time. Based on our classifier's performance, the concept of modus could be used in future human-human co-manipulation research to identify and quantify a desired behavior. Being able to quantify and identify desired behavior will allow future researchers to pick metrics to evaluate the performance of human-human teams and then program objective functions for robot controllers in human-robot teams to match or at least approach that performance.

References

[1] [n. d.]. Definition of Modus. Retrieved from https://www.merriam-webster.com/dictionary/modus

[2] Bahareh Abbasi, Natawut Monaikul, Zhanibek Rysbek, Barbara Di Eugenio, and Miloš Žefran. 2019. A multimodal human-robot interaction manager for assistive robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '19)*. IEEE, 6756–6762.

[3] Don Joven Agravante, Andrea Cherubini, Antoine Bussy, Pierre Gergondet, and Abderrahmane Kheddar. 2014. Collaborative human-humanoid carrying using vision and haptic sensing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '14)*. IEEE, 607–612.

[4] Don Joven Agravante, Andrea Cherubini, Antoine Bussy, and Abderrahmane Kheddar. 2013. Human-humanoid joint haptic table carrying task with height stabilization using vision. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 4609–4614.

[5] Don Joven Agravante, Andrea Cherubini, Alexander Sherikov, Pierre-Brice Wieber, and Abderrahmane Kheddar. 2019. Human-humanoid collaborative carrying. *IEEE Transactions on Robotics* 35, 4 (2019), 833–846.

[6] Zaid Al-Saadi, Doganay Sirintuna, Ayse Kucukyilmaz, and Cagatay Basdogan. 2020. A novel haptic feature set for the classification of interactive motor behaviors in collaborative object transfer. *IEEE Transactions on Haptics* 14, 2 (2020), 384–395.

[7] Erik Berger, David Vogt, Nooshin Haji-Ghassemi, Bernhard Jung, and Heni Ben Amor. 2013. Inferring guidance information in cooperative human-robot tasks. In *Proceedings of the 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids '13)*. IEEE, 124–129.

[8] Anastasia Bolotnikova, Sébastien Courtois, and Abderrahmane Kheddar. 2021. Adaptive task-space force control for humanoid-to-human assistance. *IEEE Robotics and Automation Letters* 6, 3 (2021), 5705–5712.

[9] Charles Bouveyron and Stéphane Girard. 2009. Robust supervised classification with mixture models: Learning from data with uncertain labels. *Pattern Recognition* 42, 11 (2009), 2649–2658. DOI: https://doi.org/10.1016/j.patcog.2009.03.027

[10] Etienne Burdet, David W. Franklin, and Theodore E. Milner. 2013. *Human Robotics: Neuromechanics and Motor Control*. The MIT Press.

[11] Antoine Bussy, Pierre Gergondet, Abderrahmane Kheddar, François Keith, and André Crosnier. 2012a. Proactive behavior of a humanoid robot in a haptic transportation task with a human partner. In *Proceedings of the 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 962–967.

[12] Antoine Bussy, Abderrahmane Kheddar, André Crosnier, and François Keith. 2012b. Human-humanoid haptic joint object transportation case study. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3633–3638.

[13] Jou-Fan Chen, Wei-Lun Chen, Chun-Ping Huang, Szu-Hao Huang, and An-Pin Chen. 2016. Financial time-series data analysis using deep convolutional neural networks. In *Proceedings of the 7th International Conference on Cloud Computing and Big Data (CCBD '16)*. IEEE, 87–92. DOI: https://doi.org/10.1109/CCBD.2016.027

[14] Sungjoon Choi, Kyungjae Lee, Sungbin Lim, and Songhwai Oh. 2018. Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '18)*. 6915–6922. DOI: https://doi.org/10.1109/ICRA.2018.8462978

[15] Brecht Corteville, Erwin Aertbeliën, Herman Bruyninckx, Joris De Schutter, and Hendrik Van Brussel. 2007. Human-inspired robot assistant for fast point-to-point movements. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*. IEEE, 3639–3644.

[16] Fotios Dimeas and Nikos Aspragathos. 2015. Reinforcement learning of variable admittance control for human-robot co-manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '15)*. IEEE, 1011–1016.

[17] Vincent Duchaine and Clement M Gosselin. 2007. General model of human-robot cooperation using a novel velocity based variable impedance control. In *Proceedings of the 2nd Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC '07)*. IEEE, 446–451.

[18] Vincent Duchaine and Clement M Gosselin. 2008. Investigation of human-robot interaction stability using Lyapunov theory. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*. IEEE, 2189–2194.

[19] Fanny Ficuciello, Luigi Villani, and Bruno Siciliano. 2015. Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *IEEE Transactions on Robotics* 31, 4 (2015), 850–863.

[20] Xiaoshan Gao, Liang Yan, Gang Wang, and Chris Gerada. 2021. Hybrid recurrent neural network architecture-based intention recognition for human-robot collaboration. *IEEE Transactions on Cybernetics* 53, 3 (2021), 1578–1586.

[21] Michael Gienger, Dirk Ruiken, Tamas Bates, Mohamed Regaieg, Michael MeiBner, Jens Kober, Philipp Seiwald, and Arne-Christoph Hildebrandt. 2018. Human-robot cooperative object manipulation with contact changes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '18)*. IEEE, 1354–1360.

[22] Raphaela Groten, Daniela Feth, Harriet Goshy, Angelika Peer, David A Kenny, and Martin Buss. 2009. Experimental analysis of dominance in haptic collaboration. In *Proceedings of the RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 723–729.

[23] Boris Ivanovic, Edward Schmerling, Karen Leung, and Marco Pavone. 2018. Generative modeling of multimodal multi-human behavior. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '18)*. 3088–3095. DOI: https://doi.org/10.1109/IROS.2018.8594393

[24] Spencer W. Jensen, John L. Salmon, and Marc D. Killpack. 2021. Trends in haptic communication of human-human dyads: Toward natural human-robot co-manipulation. *Frontiers in Neurorobotics* 15 (2021), 7.

[25] Gitae Kang, Hyun Seok Oh, Joon Kyue Seo, Uikyum Kim, and Hyouk Ryeol Choi. 2019. Variable admittance control of robot manipulators based on human intention. *IEEE/ASME Transactions on Mechatronics* 24, 3 (2019), 1023–1032.

[26] Yiannis Karayiannidis, Christian Smith, and Danica Kragic. 2014. Mapping human intentions to robot motions via physical interaction through a jointly-held object. In *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 391–397.

[27] Oussama Khatib. 1999. Mobile manipulation: The robotic assistant. *Robotics and Autonomous Systems* 26, 2–3 (1999), 175–183.

[28] Wansoo Kim, Jinoh Lee, Luka Peternel, Nikos Tsagarakis, and Arash Ajoudani. 2017. Anticipatory robot assistance for the prevention of human static joint overloading in human–robot collaboration. *IEEE Robotics and Automation Letters* 3, 1 (2017), 68–75.

[29] Kazuhiro Kosuge, Manabu Sato, and Norihide Kazamura. 2000. Mobile robot helper. In *Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Vol. 1. IEEE, 583–588.

[30] Ayse Kucukyilmaz and Illimar Issak. 2019. Online identification of interaction behaviors from haptic data during collaborative object transfer. *IEEE Robotics and Automation Letters* 5, 1 (2019), 96–102.

[31] Jessica Lanini, Hamed Razavi, Julen Urain, and Auke Ijspeert. 2018. Human intention detection as a multiclass classification problem: Application in physical human–robot interaction while walking. *IEEE Robotics and Automation Letters* 3, 4 (2018), 4171–4178.

[32] Marta Lorenzini, Wansoo Kim, Elena De Momi, and Arash Ajoudani. 2018. A synergistic approach to the real-time estimation of the feet ground reaction forces and centers of pressure in humans with application to human–robot collaboration. *IEEE Robotics and Automation Letters* 3, 4 (2018), 3654–3661.

[33] F. Landis Markley, Yang Cheng, John L. Crassidis, and Yaakov Oshman. 2007. Averaging quaternions. *Journal of Guidance, Control, and Dynamics* 30, 4 (2007), 1193–1197.

[34] José Ramón Medina, Michael Shelley, Dongheui Lee, Wataru Takano, and Sandra Hirche. 2012. Towards interactive physical robotic assistance: Parameterizing motion primitives through natural language. In *Proceedings of the 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 1097–1102.

[35] Erich Mielke, Eric Townsend, David Wingate, John L. Salmon, and Marc D. Killpack. 2024. Human-robot planar co-manipulation of extended objects: data-driven models and control from human-human dyads. *Frontiers in Neurorobotics* 18 (2024), 1291694.

[36] Erich Allen Mielke. 2018. *Force and Motion Based Methods for Planar Human-Robot Co-Manipulation of Extended Objects*. Brigham Young University.

[37] Erich A. Mielke, Eric C. Townsend, and Marc D. Killpack. 2017. Analysis of rigid extended object co-manipulation by human dyads: Lateral movement characterization. arXiv:1702.00733 Retrieved from https://www.roboticsproceedings.org/rss13/p47.pdf

[38] Keivan Mojtahedi, Qiushi Fu, and Marco Santello. 2017. On the role of physical interaction on performance of object manipulation by dyads. *Frontiers in Human Neuroscience* 11 (2017), 533.

[39] Ehsan Noohi and Milos Zefran. 2014. Quantitative measures of cooperation for a dyadic physical interaction task. In *Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 469–474.

[40] Ehsan Noohi and Miloš Žefran. 2016. Modeling the interaction force during a haptically-coupled cooperative manipulation. In *Proceedings of the 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN '16)*. IEEE, 119–124.

[41] Ehsan Noohi and Miloš Žefran. 2017. Estimating human intention during a human–robot cooperative task based on the internal force model. In *Trends in Control and Decision-Making for Human–Robot Collaboration Systems*. Yue Wang and Fumin Zhang (Eds.), Springer, 83–109. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-40533-9_5

[42] Ehsan Noohi, Miloš Žefran, and James L Patton. 2016. A model for human–human collaborative object manipulation and its application to human–robot interaction. *IEEE Transactions on Robotics* 32, 4 (2016), 880–896.

[43] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu, 2017. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics* 28, 1 (Feb. 2017), 162–169. DOI: https://doi.org/10.21629/JSEE.2017.01.18

[44] Uchenna Emeoha Ogenyi, Jinguo Liu, Chenguang Yang, Zhaojie Ju, and Honghai Liu. 2019. Physical human–robot collaboration: Robotic systems, learning methods, collaborative strategies, sensors, and actuators. *IEEE Transactions on Cybernetics* 51, 4 (2019), 1888–1901.

[45] Gerhard Pahl and Wolfgang Beitz. 2013. *Engineering Design: A Systematic Approach*. Springer Science & Business Media.

[46] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. 2009. ROS: An open-source robot operating system. In *ICRA Workshop on Open Source Software*, Vol. 3. 5.

[47] Zhanibek Rysbek, Ki Hwan Oh, Bahareh Abbasi, Miloš Žefran, and Barbara Di Eugenio. 2021. Physical action primitives for collaborative decision making in human-human manipulation. In *Proceedings of the 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN '21)*. IEEE, 1319–1325.

[48] Andrew Sawers, Tapomayukh Bhattacharjee, J. Lucas McKay, Madeleine E. Hackney, Charles C. Kemp, and Lena H Ting. 2017. Small forces that differ with prior motor experience can communicate movement goals during human-human physical interaction. *Journal of Neuroengineering and Rehabilitation* 14, 1 (2017), 1–13.

[49] Meher T Shaikh and Michael A Goodrich. 2017. Design and evaluation of adverb palette: A GUI for selecting tradeoffs in multi-objective optimization problems. In *Proceedings of the 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI '17). IEEE*, 389–397.

[50] Meher T. Shaikh and Michael A. Goodrich. 2018. When does a human replan? Exploring intent-based replanning in multi-objective path planning. In *Proceedings of the Unmanned Systems Technology XX*, Vol. 10640. International Society for Optics and Photonics, 106400G.

[51] Meher T. Shaikh and Michael A. Goodrich. 2019. Intent-based robotic path-replanning: When to adapt new paths in dynamic environments. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '19)*. IEEE, 2857–2863.

[52] Meher T. Shaikh and Michael A. Goodrich. 2020. A measure to match robot plans to human intent: A case study in multi-objective human-robot path-planning. In *Proceedings of the 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN '20)*. IEEE, 1033–1040.

[53] Meher T Shaikh, Michael A Goodrich, Daqing Yi, and Joseph Hoehne. 2016. Interactive multi-objective path planning through a palette-based user interface. In *Proceedings of the Unmanned Systems Technology XVIII*, Vol. 9837. International Society for Optics and Photonics, 98370K.

[54] Mojtaba Sharifi, Amir Zakerimanesh, Javad K. Mehr, Ali Torabi, Vivian K. Mushahwar, and Mahdi Tavakoli. 2021. Impedance variation and learning strategies in human-robot interaction. *IEEE Transactions on Cybernetics* 52, 7 (2021), 6462–6475.

[55] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556. Retrieved from https://arxiv.org/abs/1409.1556

[56] Theodoros Stouraitis, Iordanis Chatzinikolaidis, Michael Gienger, and Sethu Vijayakumar. 2018. Dyadic collaborative manipulation through hybrid trajectory optimization. In *Proceedings of the Conference on Robot Learning (CoRL)*. 869–878.

[57] Jörg Stückler and Sven Behnke. 2011. Following human guidance to cooperatively carry a large object. In *Proceedings of the 2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 218–223.

[58] Sonny Tarbouriech, Benjamin Navarro, Philippe Fraisse, André Crosnier, Andrea Cherubini, and Damien Sallé. 2019. Admittance control for collaborative dual-arm manipulation. In *Proceedings of the 19th International Conference on Advanced Robotics (ICAR '19)*. IEEE, 198–204.

[59] Eric Christopher Townsend. 2017. *Estimating Short-Term Human Intent for Physical Human-Robot Co-Manipulation*. Brigham Young University.

[60] U.S. Bureau of Labor Statistics. 2004. *Lost-Worktime Injuries and Illnesses: Characteristics and Resulting Days Away from Work*. News Release.

[61] Linda van der Spaa, Michael Gienger, Tamas Bates, and Jens Kober. 2020. Predicting and optimizing ergonomics in physical human-robot cooperation tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '20)*. IEEE, 1799–1805.

[62] Kazuhiko Yokoyama, Hiroyuki Handa, Takakatsu Isozumi, Yutaro Fukase, Kenji Kaneko, Fumio Kanehiro, Yoshihiro Kawai, Fumiaki Tomita, and Hirohisa Hirukawa. 2003. Cooperative works by a human and a humanoid robot. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, Vol. 3. IEEE, 2985–2991.

[63] Xinbo Yu, Wei He, Qing Li, Yanan Li, and Bin Li. 2021. Human-robot co-carrying using visual and force sensing. *IEEE Transactions on Industrial Electronics* 68, 9 (Sept. 2021), 8657–8666. DOI: https://doi.org/10.1109/TIE.2020.3016271

[64] Xinbo Yu, Wei He, Yanan Li, Chengqian Xue, Jianqiang Li, Jianxiao Zou, and Chenguang Yang. 2019. Bayesian estimation of human impedance and motion intention for human–robot collaboration. *IEEE Transactions on Cybernetics* 51, 4 (2019), 1822–1834.

[65] Xinbo Yu, Bin Li, Wei He, Yanghe Feng, Long Cheng, and Carlos Silvestre. 2022. Adaptive-constrained impedance control for human–robot co-transportation. *IEEE Transactions on Cybernetics* 52, 12 (Dec. 2022), 13237–13249. DOI: https://doi.org/10.1109/TCYB.2021.3107357

Received 15 February 2023; revised 14 January 2024; accepted 22 March 2024